سلسلة ملخصات شوم ١٩٩٠

نظريات ومسائل في

البرمجة بالفورتران

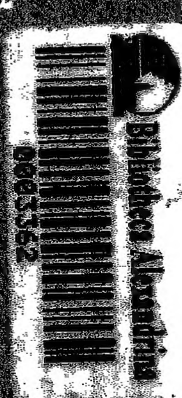
يتضمن الفورتران الهيكلي

سيمور ليبشتز

ارثر بو

يحتوي الكتاب على ٣٧٥ مسألة محلولة

سلسلة ملخصات شوم في الحاسبات



الدار الدولية للنشر والتوزيع

القاهرة – الكويت – لندن

ملخصات شوم

نظريات ومسائل

في

البرمجة بالفورتران

يتضمن الفورتران الهيكلي

٠١١٨٢

تأليف

سيمور ليبشتز Ph. D.
أستاذ الرياضيات
جامعة تمبل

أرثر بو Ph. D.
أستاذ مشارك علوم الحاسب
جامعة تمبل

ترجمة

ماجدة صلاح الدين سلامة
مخطط برامج - مركز بحوث الحسابات
العلمية والإحصائية - جامعة القاهرة
جمهورية مصر العربية

ابتسام صديق أبو الخير
ماجستير حسابات علمية
نائب مدير مركز بحوث الحسابات العلمية
والإحصائية - جامعة القاهرة
جمهورية مصر العربية

مراجعة

الأستاذ الدكتور أحمد عزيز كمال
أستاذ بكلية الهندسة - جامعة القاهرة
جمهورية مصر العربية



الدار الدولية للنشر والتوزيع

القاهرة - الكويت - لندن

حقوق النشر

Programming with Fortran
Including Structured Fortran
Seymour Lipschutz
Arthur Poe

الدار الدولية للنشر والتوزيع

٣٨ ش الأهرام – روكسى – مصر الجديدة

ص . ب : ٥٥٩٩ هليوبوليس غرب – القاهرة

ت : ٢٥٨٢٨٨٧ – تلكس PBCRB ٢٠٠٧٠

فاكس . ٢٩١٨٠٥٩ / ٢٠٢ ..

ISBN 07 084821

بسم الله الرحمن الرحيم

مقدمة الناشر

المعرفة هى أصل الحضارة ،
والكلمة هى مصدر المعرفة ،
والكلمة المطبوعة هى أهم مكون فى هذا المصدر .

وقد كانت الكلمة المطبوعة ولاتزال أهم وسائل الثقافة والاعلام وأوسعها انتشاراً وأبقاها أثراً ، حيث حملت إلينا حضارات الأمم عبر آلاف السنين لتتولى الأجيال المتلاحقة صياغة حضاراتها وإضاءة الطريق بنور العلم والمعرفة .

والكلمة تبقى مجرد فكرة لدى صاحبها حتى تتاح لها فرصة نشرها وترجمتها إلى لغات الآخرين ثم توزيعها ، وذلك وحده هو الذى يكفل لها أداء رسالتها .

وعالم الكتب العلمية عالم رحب ممتد الآفاق ، متسع الجنبات ، والعلم لا وطن له ولا حدود ، ويوم يحظى القارىء بأحدث الكتب العلمية باللغة العربية لهو اليوم الذى تتطلع له الأمة العربية جمعاء .

والدار الدولية للنشر والتوزيع تشعر بالرضا عن مساهمتها فى هذا المجال بتقديم الطبعات العربية للكتب العلمية الصادرة عن دار ماكجروهيل للنشر بموجب الاتفاق المبرم معها ، مستهدفة توفير احتياجات القارىء العربى أستاذاً وباحثاً وممارساً .

ومن جانب آخر فنحن نمد يدنا إلى الجامعات العربية والمراكز العلمية والمؤسسات والهيئات الثقافية للتعاون معنا فى إصدار طبعات عربية حديثة من الكتب والمراجع العلمية تخدم التقدم العلمى والحضارى للقارىء العربى .

والله ولى التوفيق

محمد وفائى كامل
مدير عام
الدار الدولية للنشر والتوزيع



تقديم

يمكن أن يطلق على النصف الثانى من القرن العشرين عهد الحاسبات . فالناس فى جميع مجالات العمل تقريباً (مثل الهندسة ، العلوم الطبيعية ، الاقتصاد ، علم النفس ، التعليم ، العلوم الاجتماعية ، العلوم الطبية ، القانون والأعمال ، أى فى كل مجال يحتاج لجمع البيانات وتحليلها بواسطة الحاسبات سيكون لهم اتصال ما بالحاسبات واللغات المستعملة فى توجيهها . وتسمى إحدى هذه اللغات فورتران FORTRAN وهى اختصار FOR mula TRANslation . (أى ترجمة الصيغ الرياضية) . وقد تم تصميم وكتابة هذا الكتاب لتقديم لغة الفورتران ولتعليم طرق حل المسائل باستخدام هذه اللغة . والهدف الأساسى لهذا الكتاب هو تعليم القارىء كيف يكتب برامج فورتران واضحة وفعالة بواسطة التركيز على الأساليب الفنية والخبرات الجيدة للبرمجة وذلك بالإضافة إلى تقديم قواعد الفورتران . يناقش هذا الكتاب كل المبادىء الهامة للفورتران القياسى وكذلك بعض السمات ككتاب دراسى لمقرر تعريفى للبرمجة بالفورتران ، أو كمكمل للكتب الدراسية المعتادة فى المادة الدراسية « مقدمة لعلم الحاسبات الآلية » .

سيروق هذا الكتاب لعدد كبير من القراء ، كما أنه سيكون بمثابة دليل فعال للتعليم الذاتى ، ويرجع ذلك لمنهجه المبسط وكذلك تدرج أمثلته . يبدأ كل فصل بجملة واضحة عن التعاريف والأساسيات المتعلقة بالموضوع مع مادة توضيحية ووصفية . يلى ذلك مجموعة متدرجة من المسائل المحلولة والتكميلية . فالمسائل المحلولة تستخدم فى توضيح وتقوية المادة بينما تقدم المسائل التكميلية مراجعة كاملة للمادة المقدمة فى الفصل .

ينقسم الكتاب إلى اثنى عشر فصلا . يتناول الفصل الأول باختصار العمليات الأساسية للحاسب . وذلك من خلال استخدام برامج بسيطة وهذا يعطى القارىء لمحة عن لغة الفورتران وكذلك بعض الشعور بالديناميكية المتضمنة فى البرمجة بالفورتران . ويناقش أيضاً هذا الفصل تعليمات تثقيب وتنظيم حزمة الفورتران حتى يتمكن القارىء من كتابة وتشغيل بعض البرامج البسيطة من بداية تعلمه اللغة . يبدأ التقديم الرسمى للغة الفورتران ، فى الفصل الثانى ، وفيه تقدم التعبيرات الرياضية وجمل التخصيص الحسابية وتناقش أيضاً حسابات الأعداد الصحيحة والحقيقية ، وكذلك بعض العثرات التى تتعرض لها الحسابات بالحاسب . يتعلق الفصل الثالث بعمليات الإدخال / الإخراج العددية . وكذلك يناقش كل من الملامح المصاغة وغير المصاغة .

يقدم الفصل الرابع طريقة رسم خرائط سير العمليات وذلك لمساعدة القارىء فى وضع وتصور خطوات حل المسائل . وأهم من ذلك ، فالفصل الرابع يناقش جمل نقل التحكم فى الفورتران الأساسى (غير الهيكل) بما فى ذلك جملة IF المنطقية . ويناقش الفصل الثانى عشر ملامح التحكم الهيكل ، بما فى ذلك كتابة جملة IF . بالمثل يقدم الفصل الخامس حلقة DO الأساسية المفهرسة . بينما يتم معالجة حلقة DO المعممة وهياكل WHILE و FOR فى الفصل الثانى عشر . ويناقش الفصل السادس المجموعات المتراصة والمتغيرات ذات الأدلة . ولمساعدة القارىء على كتابة برامج معقدة فى وحدات قائمة بذاتها ، يقدم الفصل السابع الدوال FUNCTIONS والبرامج الفرعية SUBROUTINE .

يعالج الفصل الثامن أساليب البرمجة للبحث والفرز وصيانة الملفات الخ . كما تناقش الحسابات العددية مثل إيجاد أصفار الدوال وحل نظم المعادلات الخطية . وتعالج معلومات الحروف والمتغيرات المنطقية مع الإدخال / الإخراج فى الفصل التاسع . يغطى الفصل العاشر ملامح إضافية من الإدخال / الإخراج مثل حقل — G وصيغة وقت التنفيذ . يقدم الفصل الحادى عشر ملامح أخرى للفورتران

مثـــل متغيرات DOUBLE PRECISION و COMPLEX وجمل COMMON و EQUIVALENCE . وكما سبق أن ذكرنا ، فالفصل الثانى عشر يناقش ملامح الفورتران الهيكلى مثل كتابة جملة IF وحلقة DO المعممة . وهذا الفصل متوافق مع المتغيرات الحديثة فى الفورتران القياسى ومع مترجمات WATFOR و WATFIV .

كتبت المواضيع والفصول بطريقة تجعلها تكاد تكون مستقلة بعضها عن بعض . وذلك بغرض جعل الكتاب أكثر مرونة ويستفاد منه كمرجع . وقد ضم أيضاً ملحقاً للتمثيل الداخلى للبيانات فى الحاسب .

نود أن نشكر الكثيرين من الأصدقاء والزملاء على المقترحات القيمة والمراجعة الدقيقة لأصول الكتاب . نود أيضاً أن نعبر عن امتناننا لأعضاء هيئة ماكجروهيل لسلسلة تشوم وعلى الأخص جون اليـــانو والين لابربيرا لمعاونتهما الدائبة .

سيمور ليبشتز

أرثر ت . بو

المحتويات

الصفحة

الفصل الأول

المقدمة وتنظيم البرنامج

١ ــ ١ مقدمــــة

يمكن تسمية النصف الثانى من القرن العشرين بعصر الحاسبات الآلية ولقد انتشرت هذه الآلات و « اللغات » المتعددة المستخدمة فى توجيهها حتى أصبحت تقارب انتشار الآلات الكاتبة والمساطر الحاسبة . يقوم الحاسب الآلى أساساً بتنفيذ ثلاث وظائف :

١ ــ استقبال بيانات (إدخال)

٢ ــ تشغيل البيانات بحسابات متعددة

٣ ــ إصدار بيــانات (إخراج)

تتم هذه الوظائف بإعطاء الحاسب برنامجاً يتكون من سلسلة من الجمل (تسمى مجموعة من الجمل المتتالية جزءاً من البرنامج) .

والهدف الأساسى من هذا الكتاب هوتعليم القارىء كيف يكتب برنامج لحاسب باستخدام لغة الفورتران . وكلمة فورتران FORTRAN مأخوذة من FORmula TRANslation أى ترجمة الصيغ الرياضية) وستبدأ هذه الدراسة رسمياً من الفصل التالى . سيغطى هذا الفصل بعض الاعتبارات الخاصة بتثقيب جمل الفورتران على البطاقات وكيفية إعداد حزمة الفورتران لحاسب . وسنتعلم أيضاً بعض الحقائق عن الحاسبات والتى يمكن أن تعطينا إحساساً بالديناميكية المتضمنة فى البرمجة بلغة الفورتران . وسنرى بالتحديد كيف يقوم الحاسب بتنفيذ بعض البرامج البسيطة . (وإننا لنحبذ أن يقوم القــارىء فعلا بتثقيب برنامج أو أكثر من هذه البرامج على بطاقات وتشغيلها على الحاسب) .

وسنختتم الفصل بنظرة شاملة على الحاسبات ولغاتها . وتشتمل هذه النظرة على تفاصيل كثيرة لن نحتاجها على الإطلاق فى كثير من المواضيع اللاحقة فى هذا الكتاب . ولذلك يمكن أن يمر القارىء على هذه الأجزاء مرورا سريعاً فى أول قراءة لهذا الكتاب وأن يرجع إليها عند الضرورة .

١ ــ ٢ تثقيب جمل الفورتران

أحياناً نبدأ بكتابة برامج الفورتران على صحائف تكويد (أنظر شكل ١ ــ ١) . إلا أنه بمجرد كتابتها يجب إعطاؤها لحاسب بشكل يمكن للآلة أن تقرأه مثل مجموعة بطاقات مثقبة أو شريط ورق أو شريط ممغنط ، أو من خلال آلة كاتبة طرفية وذلك يتوقف على وحدة الإدخال . وسنفترض خلال هذا الكتاب أن وحدة الإدخال الخاصة بنا هى وحدة قراءة البطاقات المثقبة ، وعلى ذلك فسوف تثقب البرامج على بطاقات . تحتوى هذه البطاقة على 80 عموداً كما فى الشكل ١ ــ ٢ (أ) . بحيث يمكن أن يتسع كل عمود لحرف واحد .

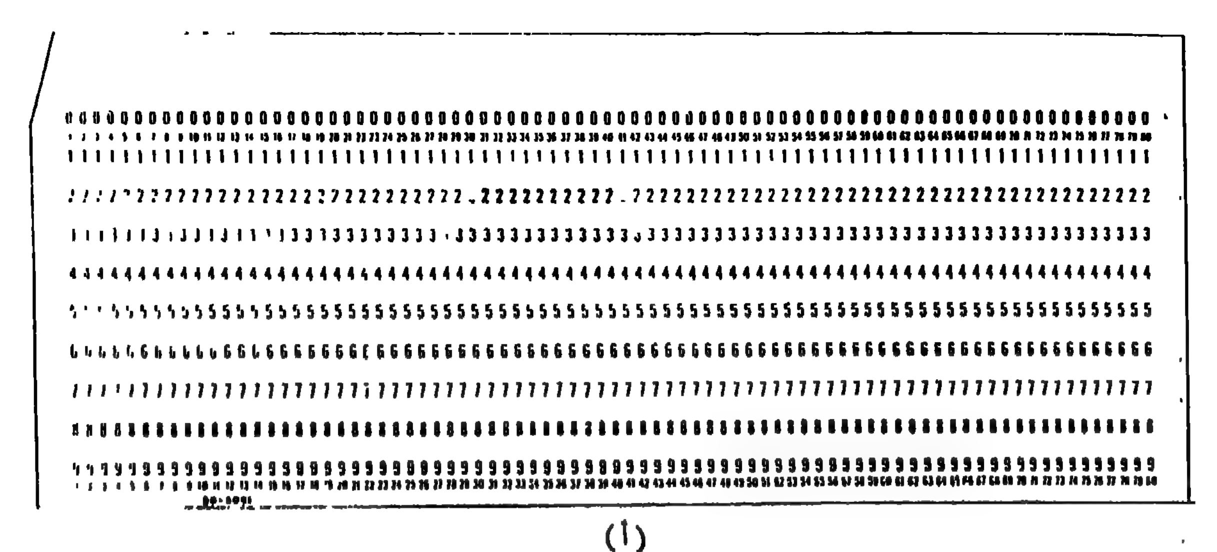
يجب أن تثقب كل جملة فى برنامج الفورتران على بطاقة منفصلة . وكما هو موضح فى شكل ١ ــ ٢ (ب) ، ليس من الضرورى أن تستخدم جملة الفورتران جميع الأعمدة الثمانين بالبطاقة . بالتحديد لدينا القوانين التالية :

١ ــ تثقب جمل الفورتران الأصلية من العمود 7 إلى العمود 72 فقط (شاملا 7 و 72)

STATEMENT NUMBER | FORTRAN STATEMENT → | IDENT

1 5 6 7 10 20 30 40 50 60 72 73 80

شكل ١ – ١ صحيفة تكويد الفورتران



(أ)

1 5 6 7 72 80

رمز التكملة

التعليق C ← عنوان الجملة

← جملة الفورتران →

رقم البطاقة أو أى تعريف آخر

(ب)

شكل ١ – ٢

٠ – يسمى عمود 6 عمود التكملة ويدخر للفرض التالى : إذا ماكانت جملة الفورتران أطول من أعمدة البطاقة وعددها 66 كما ذكرنا سابقا ، فيمكن تكملتها حينئذ فى بطاقات أخرى (بين الأعمدة 7 إلى 72) بما لايزيد عن 19 بطاقة إضافية . فى هذه الحالة يجب أن يثقب « 1 » (أو أى علامة أخرى غير الصفر) فى العمود 6 من كل بطاقة من البطاقات الإضافية . بمعنى أن أى علامة (غير الصفر) مثقبة فى العمود 6 تدل الحاسب أن هذه الجملة هى تكملة للجملة السابقة .

٣ – يمكن وضع عناوين للاستدلال عليها ، وهذه العناوين تثقب من العمود الأول إلى العمود الخامس من بطاقة الجملة المعنونة . والعنوان فى الفورتر . هو أى رقم صحيح موجب بدون إشارة جبرية (بحد أقصى خمس خانات) ويسمى رقم الجملة . ومن الواضح أنه لايجوز أن يتكرر رقم الجملة لجملتين مختلفتين .

٤ – يتجاهل الحاسب محتويات الأعمدة إبتداء من العمود 73 إلى 80 . ومن الممكن تثقيب هذه الأعمدة للتعريف أو التسلسل أو لأى غرض آخر .

يمكن إضافة تعليقات إلى البرنامج بتثقيب « C » فى العمود الأول من البطاقة . ويثقب التعليق نفسه فى أى مكان بين العمود 2 والعمود 80 . ولاتمثل التعليقات أى عبء حقيقى على البرنامج . فهى تستخدم عادة لتعريف البرنامج وللإيضاح ، وببساطة يتخطى الحاسب التعليق ويتحرك إلى الجملة التالية إلا أن التعليق سوف يظهر فى صفحة الطباعة حين يقوم الحاسب بطباعتها .

ملحوظة : سوف يلاحظ القارىء عند تثقيب البطاقات أن الحروف الكبيرة فقط هى المتاحة بالنسبة لحروف الأبجدية . تكتب علامات الجمع ، الطرح ، الضرب والقسمة كالآتى + ، – ، * ، / على الترتيب . أما ** فترمز إلى الأس ، أى رفع الرقم إلى قوة ما (هذه العمليات ستناقش بالتفصيل فى الفصل الثانى) وعلى هذا فى الفورتران :

$a.b$ تكتب A*B و $\frac{a}{b}$ تكتب A/B و a^b تكتب A**B

ونلاحظ أن الحاسب فى الفورتران يتجاهل المسافات الخالية وعلى ذلك يمكن القارىء إضافتها لتسهيل عملية القراءة . فعلى سبيل المثال يمكن تثقيب

A**2 + B**2 بدلا مايكافئه فى A**2+B**2

و 4*I + 6*J بدلا مايكافئه فى 4*I+6*J

و GO TO بدلا من GOTO

١ – ٣ حزمة الفورتران

برنامج الفورتران الذى نكتبه ونثقبه على بطاقات هو مجرد جزء واحد من الحزمة التى نعطيها للحاسب إذا أريد تنفيذ البرنامج . وتتكون هذه الحزمة من سلسلة البطاقات التالية :

١ – بطاقة تعريف تحتوى هذه البطاقة عادة على اسم مخطط البرنامج ، المشروع ، المادة الدراسية أو رقم الحساب ، رقم الأولوية الحد الأقصى لوقت التشغيل ، الخ .

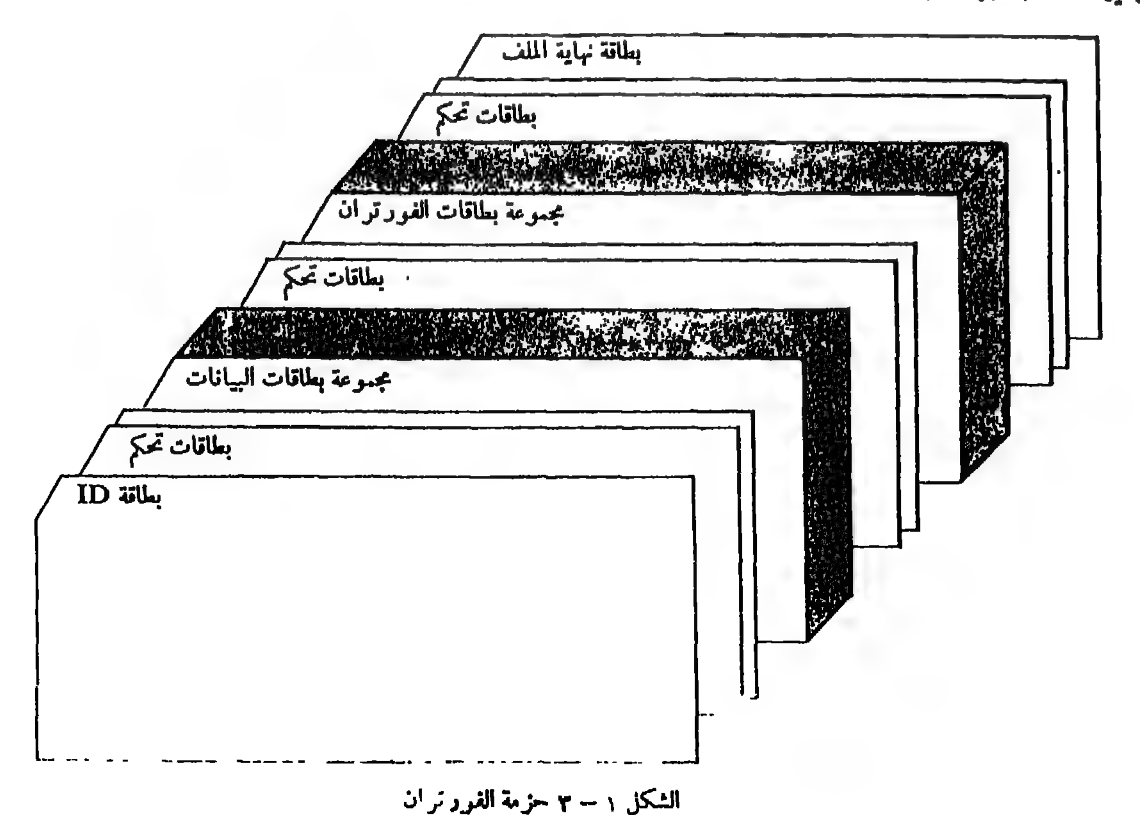
٢ – مجموعة بطاقات الفورتران ، تحتوى هذه المجموعة من البطاقات على برنامج الفورتران الفعلى .

٣ – مجموعة بطاقات البيانات ، تحتوى هذه المجموعة من البطاقات ، والتى قد تكون خالية ، على البيانات التى تقرأ بواسطة الحاسب .

٤ – بطاقة نهاية الملف ، تحدد هذه البطاقة نهاية الحزمة .

من الممكن فصل المجموعات الأربع السابقة بواسطة بطاقة أو أكثر من بطاقات التحكم كما هو موضح فى الشكل ١ – ٣ . والمعلومات الفعلية التى تحتويها بطاقة التعريف وبطاقة نهاية الملف وبطاقة التحكم تعتمد على الحاسب المستخدم .

تحتوى مجموعة بطاقات البيانات على المدخلات ، من الممكن إستخدام كل أعمدة البطاقة الثمانين عند تثقيب بطاقات البيانات ، أى أن القوانين الخاصة بتثقيب جمل الفورتران لا تطبق فى هذه الحالة ، وليس لأى عمود (العمود 6 مثلا) ، أى غرض خاص .



الشكل ١ – ٣ حزمة الفورتران

١ – ٤ تخزين الاعداد

تتكون وحدة ذاكرة الحاسب من عدة خلايا ذاكرة (أماكن) لتخزين البيانات . ولايخصص مكان فى الذاكرة لعدد إلا إذا أعطى أسما لهذا المكان . ويمثل تخصيص عدد ، مثل 75 لمكان فى الذاكرة يسمى N بواسطة

N ← 75

وبصورة عامة :

NAME ← تعبير حسابى

يدل على أن قيمة التعبير الحسابى يعطى لمكان الذاكرة المسمى NAME .

ونؤكد أن السهم ← مقبول عموماً للدلالة على « التخصيص » . إلا أن لغات البرمجة المختلفة تمثل ← بطرق مختلفة . فى الفورتران يمثل السهم بـ = (أى علامة التساوى الحسابية) . وعلى ذلك فإن جملة الفورتران :

I = 7

تأمر الحاسب بتحديد مكان الذاكرة المسمى I بالقيمة 7 . وجملة الفورتران :

J = 4 + 5 + 8

J = 4 + 5 + 8

تعطى التعليمات للحاسب ليقوم :

١ – بجمع القيم الموجودة على الجانب الأيمن من علامة = .
٢ – تخزين الناتج ، وهو 17 ، فى المكان المسمى J .

لاحظ تضمين إجراءين : إيجاد قيمة التعبير الرياضى على يمين العلامة = وتخزين الناتج فى المكان المعطى على يسار العلامة = .

يمكن أن يحتوى التعبير الرياضى أيضاً على أسماء لأماكن تخزين إلا أن هذه الأسماء يجب أن تعرف بمعنى أنها أسماء لأماكن تخزين فى الذاكرة تم فيها تخزين أرقام مسبقاً فى نفس البرنامج . يوجد الحاسب قيمة التعبير الرياضى باستبدال الأسماء بالقيم المخزنة فى أماكن التخزين هذه .

وعلى سبيل المثال أنظر جملة الفورتران :

K = 3*I + 2*J − 24

حيث تحتوى I و J على 7 ، 17 على الترتيب (تذكر أن النجمة * تعنى الضرب فى الفورتران) . يوجد الحاسب قيمة التعبير الرياضى على اليمين كالتالى :

$3 \cdot 7 + 2 \cdot 17 - 24$

ويخزن الناتج ، 31 ، فى مكان الذاكرة المسمى K . لاحظ أن محتويات I و J لاتتحرك من الأماكن I و J حين يحسب الحاسب قيمة 3*I + 2*J − 24 بل تنسخ القيم فى الوحدة الحسابية ، وتحتفظ I و J بنفس قيمها .

وسنوضح هذه الأفكار فى برنامجنا الأول .

البرنامج الأول

نكتب برنامجاً لحساب المجموع LS وحاصل الضرب LP للأرقام I = 24 : J = 3 ، K = 7 شكل ١ – ٤ (أ) هو صحيفة التكويد لبرنامج الفورتران الذى يقوم بهذه الحسابات ، وشكل ١ – ٤ (ب) يعطى صورة البرنامج بعد تثقيبه على مجموعة من البطاقات .

لاحظ أن كل جملة مكتوبة على سطر منفصل فى صحيفة التكويد ومثقبة على بطاقة منفصلة . وسنتابع الحاسب أثناء قراءته وتنفيذه البرنامج وإليك أولا بعض التفسيرات لعدة جمل فى البرنامج :

١ – أول بطاقة تحتوى على الحرف « C » مثقب فى العمود الأول ، ولذلك فهى بطاقة تعليق . كما تم توضيح ذلك من قبل فالتعليقات لا تؤثر على البرنامج ، فعندما يقابل الحاسب الحرف « C » فى العمود الأول فإنه ببساطة يقوم بطباعة التعليق ويمضى إلى الجملة التالية .

٢ – جملة PRINT

PRINT, I, J, K, LS, LP

تأمر الحاسب بطباعة المحتويات الحالية لأماكن الذاكرة I و J و K و LS و LP بهذا الترتيب . هذه جملة إخراج غير مصاغة . وتطبع القيم فى صيغة محددة مسبقاً وسندرس بالتفصيل الجمل المصاغة وغير المصاغة فى الفصل الثالث .

"C" FOR COMMENT

STATEMENT NUMBER 1 5	6	FORTRAN STATEMENT → 7 10 20 30 40 50
C		PROGRAM WHICH CALCULATES SUM AND PRODUCT
		I = 24
		J = 3
		K = 7
		LS = I+J+K
		LP = I*J*K
		PRINT, I,J,K,LS,LP
		STOP
		END

(أ)

(ب)

شكل ١ – ٤

جملة STOP تأمر الحاسب بإيقاف التنفيذ ، وجملة END هي دائماً آخر جملة في أي برنامج . (أنظر قسم ١ – ٨ . (حـ) لمزيد من التفسير عن الفرق بينهما) .

تنفيـذ البرنـامج

السطر الأول : هذا تعليق لوجود « C » مطبوعة في العمود الأول . وكما سبق ذكره فالحاسب يتخطى هذه الجملة ويمضي إلى الجملة التالية .

السطر الثاني : يأمر الحاسب بتخزين الرقم الـ ح 24 في مكان الذاكرة المسمى I :

I ← 24

السطر الثالث : J ← 3

السطر الرابع : K → 7

السطر الخامس : يأمر الحاسب بإيجاد قيمة التعبير الرياضي على يمين علامة التساوي بجمع 7 + 3 + 24 ثم تخزين الناتج ، وهو34 ، في مكان الذاكرة المسمى LS .

LS ← 34

السطر السادس : يأمر الحاسب بإيجاد قيمة التعبير الرياضي على اليمين ، بضرب 24 في 3 في 7 ، وتخزين الناتج في LP :

LP ← 504

السطر السابع : يأمر الحاسب بطباعة القيم المخزنة في I و J و K و LS و LP بهذا الترتيب ، ونمثل هذا بإحاطة الخرج بمتوازي أضلاع كالتالي :

24, 3, 7, 34, 504

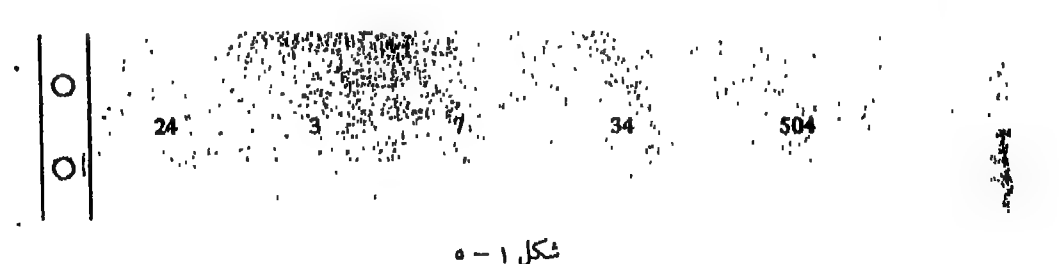
(انظر إلى التعليق في آخر هذا القسم على جملة PRINT غير المصاغة هذه) .

السطر الثامن : تأمر الحاسب بإيقاف التنفيذ .

السطر التاسع : يجب أن ينتهي كل برنامج بجملة END .

الخـــرج

سنعتبر أن وحدة الإخراج هي آلة الطباعة . وسنعتبر أن ورق الطباعة الخاص بآلة الطباعة يحتوي السطر منها على 132 عموداً (مسافات الطباعة) . وتستخدم عادة جملة الصياغة FORMAT لتوجيه الحاسب أن يضع الخرج على ورق الطباعة . في هذا القسم سنستخدم جملة PRINT غير المصاغة ، وعلى هذا فهناك صيغة محددة مسبقاً للخرج فسيظهر الخرج على سطر واحد كما في شكل ١ – ٥ .



شكل ١ – ٥

تعليق : إذا كان القارىء سينفذ البرنامج على حاسب لايقبل جملة PRINT غير المصاغة فيجب على القارىء أن يستبدل جملة PRINT (السطر السابع) بالجملتين التاليتين :

```
      WRITE(6, 31) I, J, K, LS, LP
31    FORMAT(5I12)
```

(وسوف يتم شرح هذا الثنائى WRITE-FORMAT فى الفصل الثالث)

١ ـــ ٥ قراءة البيانات

قام البرنامج السابق بتوليد بياناته خلال التنفيذ بتخصيص قيم K, J, I صراحة . من الممكن أيضاً أن « يقرأ » الحاسب بطاقات البيانات والشرائط الممغنطة أو أى وحدة إدخال أخرى . وحيث أننا افترضنا استعمال وحدة قراءة البطاقات ، فإن الحاسب سيقرأ البيانات من مجموعة بطاقات . (وكما هو موضح فى شكل ١ — ٣ ، تظهر مجموعة بطاقات البيانات بعد مجموعة بطاقات برنامج الفورتران فى حزمة الفورتران المعطاة للحاسب) .

وتعطى التعليمات للحاسب بقراءة البيانات من بطاقات البيانات باستخدام جملة READ . وتبدو جملة READ غير المصاغة على النحو التالى :

```
READ, I, J, K, M, N
```

لاحظ أن كلمة READ وكل متغير ، فيما عدا الأخير ، متبوعة بفصلة (,) هذه الجملة تعطى تعليمات للحاسب بتخزين القيم الخمس الأولى من مجموعة البيانات فى الأماكن المسماة I ، J ، K ، M ، N على الترتيب

والميزة الأساسية لاستخدام جملة READ هى أن تجعل البرنامج بصورة أعم ، مستقل عن بيانات الإدخال . على سبيل المثال ، لو أننا استخدمنا جملة READ فى البرنامج السابق لقراءة قيم للأماكن I ، J ، K لكان من الممكن استخدام هذا البرنامج لإيجاد مجموع وحاصل ضرب أى قيم I ، J ، K وليس فقط القيم ٢٤ ، 3 ، 7 .

مثال ١ — ١

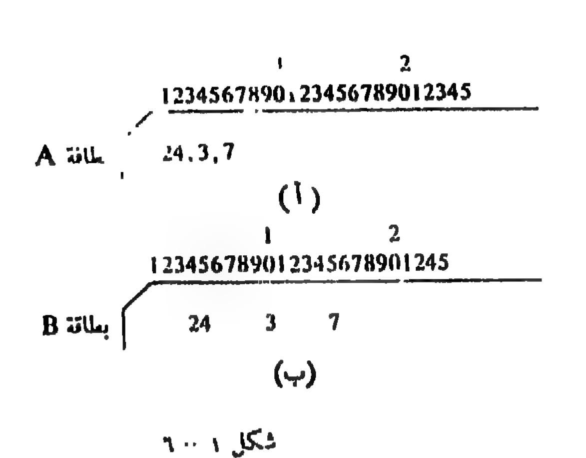
(أ) سنعيد كتابة البرنامج السابق إلا أننا سوف نقرأ قيم I ، J ، K . بالتحديد سوف نستبدل الجمل الثلاث بعد التعليق بالجملة الواحدة .

```
READ, I, J, K
```

وتثقب بطاقة مثل بطاقة A فى الشكل ١—٦ (أ) . فإذا لم يكن متاحاً القارىء استخدام إدخال وإخراج غير مصاغ فعليه استخدام الجملتين :

```
      READ(5, 10) I, J, K
10    FORMAT(3I5)
```

ويثقب بطاقة بيانات بديلة للسابقة كما هو موضح فى البطاقة B فى الشكل ١ — ٦ (ب) . سيكون خرج هذا البرنامج هو نفس خرج البرنامج الأول .



شكل ١ — ٦

(ب) إشرح الغرض من البرنامج

```
READ, A, B, C, D, E
SUM = A + B + C + D + E
PRINT, A, B, C, D, E
PRINT, SUM
STOP
END
```

هذا البرنامج يحسب مجموع خمسة أرقام وتوجد هنا جملتا PRINT . قيم A ، B ، C ، D ستطبع على سطر ويطبع مجموعها (SUM) على سطر آخر .

(حـ) إشرح الغرض من البرنامج :

```
READ, B, H
AREA = (B*H)/2.0
PRINT, B, H, AREA
STOP
END
```

هذا البرنامج يحسب مساحة (AREA) المثلث ذو القاعدة B والإرتفاع H . وستطبع AREA, H, B على سطر واحد .

١ ــ ٦ اتخاذ القرارات

ينفذ الحاسب عادة جمل البرنامج واحدة تلو الأخرى كما فى البرنامج السابق إلا أنه فى بعض الأحيان نود أن يكرر أو يتخطى الحاسب جزءا من البرنامج ، أى أننا نريد الحاسب أن ينتقل من نقطة فى البرنامج إلى نقطة أخرى ويمكن تنفيذ ذلك باستخدام جملة GO TO والتى لها الشكل :

GO TO *n*

حيث *n* هو « رقم جملة » . فهذه الجملة تأمر الحاسب أن ينقل التحكم إلى الجملة المعنونة *n* ، بمعنى أن الجملة التالية التى نريد تنفيذها هى الجملة التى رقمها *n* .

من أهم صفات الحاسب مقدرته على مقارنة القيم واتخاذ القرارات بناء على ذلك . على سبيل المثال ، أنظر جملة الفورتران :

```
IF(N.LT.300) GO TO 100
```

(الرمز " LT. " فى الفورتران يقابل « أقل من » فجملة IF هذه تعطى للحاسب التعليمات كما يلى :

إذا تحقق الشرط $n < 300$ اذهب لتنفيذ الجملة التى رقمها 100 ، وأن لم يتحق ذلك استكمل بتنفيذ الجملة التالية لجملة IF .

بمعنى آخر ، تسأل الجملة السؤال « IS N < 300? » . إذا كان الجواب على ذلك « نعم » ، ينقل الحاسب التحكم إلى الجملة التى رقمها 100 ، أما إذا كان الجواب « لا » يستمر الحاسب فى تنفيذ الجملة التالية .

يوضح البرنامج التالى هذه الأفكار (وستدرس هذه الأفكار بالتفصيل فى الفصل الرابع) . ويعطى البرنامج للحاسب التعليمات بطباعة الأرقام فردية الصحيحة الموجبة من 1 إلى 11 وكذلك مربعاتها .

```
C       PROGRAM ODD INTEGERS AND THEIR SQUARES
        N = 1
   21   K = N*N
        PRINT, N, K
        N = N + 2
        IF(N.LT.12) GO TO 21
        STOP
        END
```

قبل مناقشة تنفيذ البرنامج ، نلاحظ أن الجملة

$$N = N + 2$$

قد تبدو محيرة ، إلا أننا عرفنا من القسم ١ – ٤ أن علامة = فى الفورتران لا تعنى التساوى ولكن تعنى تخصيص (والتى تتضمن إجراءين) أولا يوجد الحاسب قيمة $N + 2$ باستعمال القيمة الحالية لـ N وبعد ذلك يخزن هذا الرقم الجديد فى المكان N . بمعنى آخر أن الجملة $N = N+2$ تعنى أن القيمة الجديدة لـ N هى القيمة الحالية مضافاً إليها 2 .

تنفيذ البرنـــامج

الخطوة :

١ – السطر الأول : عبارة عن تعليق حيث توجد C مثقبة فى العمود الأول .

٢ – السطر الثـــانى :

$$N \leftarrow 1$$

٣ – السطر الثالث : يأمر الحاسب بضرب قيمة N فى نفسها ، وتخزين النتيجة فى K :

$$K \leftarrow 1$$

٤ – السطر الرابع : تطبع قـيم N و K

1, 1

٥ – السطر الخامس : يأمر الحاسب بجمع 2 على القيمة الحالية لـ N ، وتخزين هذا المجموع (3) فى المكان N (بهذه الطريقة تمسح القيمة السابقة لـ N) . وتمثل حقيقة مسح القيمة 1 الموجودة فى N واستبدالها بالقيمة 3 بواسطة

$$N \leftarrow \not{1}\ 3$$

٦ – السطر السادس : يجرى السؤال التالى :

هل N أقل من 12 ؟

وحيث أن $N = 3$ ، فالجواب هو نعم ، أى أن التعبير الرياضى بين القوسين صحيح لذلك يذهب الحاسب إلى الجملة التى رقمها 21 ، أى إلى السطر الثالث من البرنامج .

٧ – السطر الثالث : يربع القيمة الحالية لـ N ويخزن الناتج (9) فى المكان K (وعلى ذلك تمسح القيمة السابقة لـ K) .

$$K \leftarrow \not{1}\ 9$$

٨ – السطر الرابع : يطبع القيم الحالية لـ N و K .

3, 9

٩ – السطر الخامس : يجمع الحاسب 2 على القيمة الحالية لـ N ، وعلى ذلك تصبح

$$N \leftarrow \not{1}\ \not{3}\ 5$$

١٠ – السطر السادس : يعود الحاسب إلى الجملة التى رقمها 21 ، أى إلى السطر الثالث فى البرنامج حيث أن الإجابة على السؤال « هل N أقل من 12 ؟ » هو نعم .

١١ – السطر الثالث : يربع القيمة الحالية لمحتويات N ويخزن الناتج (وهو 25) فى K

K ← ~~1~~ ~~9~~ 25

١٢ – السطر الرابع : يطبع القيـم الحالية لمحتويات N و K .

5, 25

١٣ – السطر الخامس : يزيد الحاسب قيمة N بمقدار 2

N ← ~~1~~ ~~3~~ ~~5~~ 7

١٤ – السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن 12 > 7 .

١٥ – السطر الثالث : يربع الحاسب N ويخزن القيمة فى K

K ← ~~1~~ ~~9~~ ~~25~~ 49

١٦ – السطر الرابع : يطبع الحاسب قيم N و K .

7, 49

١٧ – السطر الخامس : يزيد الحاسب قيمة N بمقدار 2 .

N ← ~~1~~ ~~3~~ ~~5~~ ~~7~~ 9

١٨ – السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن 12 > 9 .

١٩ – السطر الثالث : يربع الحاسب N ويخزن هذا العدد الصحيح فى K .

K ← ~~1~~ ~~9~~ ~~25~~ ~~49~~ 81

٢٠ – السطر الرابع : يطبع الحاسب قيم N و K .

9, 81

٢١ – السطر الخامس : تزداد قيمة N بمقدار 2

N ← ~~1~~ ~~3~~ ~~5~~ ~~7~~ ~~9~~ 11

٢٢ – السطر السادس : يعود الحاسب إلى السطر الثالث حيث أن 12 > 11 .

٢٣ – السطر الثالث : تستبدل قيمة K الحالية بمربع N .

K ← ~~1~~ ~~9~~ ~~25~~ ~~49~~ ~~81~~ 121

٢٤ – السطر الرابع : تطبع قيم N و K .

| 11, 121 |

٢٥ – السطر الخامس : تزداد قيمة N بمقدار 2

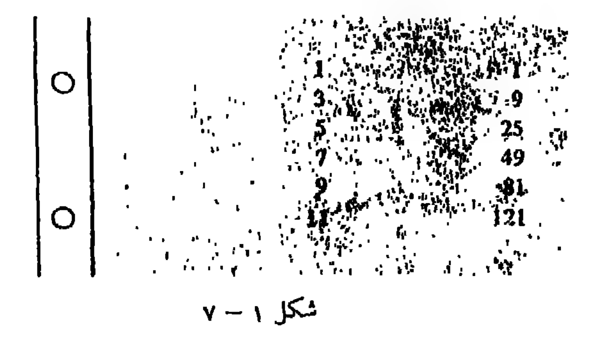
N ← ~~1~~ ~~3~~ ~~5~~ ~~7~~ ~~9~~ ~~11~~ 13

٢٦ – السطر السادس : ينفذ الحاسب الجملة التالية فى البرنامج . حيث أن جواب السؤال هل N < 12 الآن هو لا ، أى أن التعبير الرياضى بين القوسين غير صحيح ..

٢٧ – السطر الرابع : يأمر الحاسب بالتوقف STOP .

الخـــرج

فى كل مرة تظهر جملة PRINT ، يطبع الحاسب الخرج فى سطر واحد ومن ثم سيظهر كل عدد فردى ومربعه على سطر مستقل ، كما فى الشكل ١ – ٧ .



شكل ١ – ٧

تعليق ١ : لو استبدلنا السطر السادس فى البرنامج ، بما يلى مثلا :

```
IF(N.LT.100) GO TO 21
```

فإن الحاسب سيطبع الأعداد الصحيحة الفردية ومربعاتها من 1 إلى 99 .

تعليق ٢ : إذا كان القارىء سينفذ برنامجاً على حاسب لايقبل جملة PRINT غير المصاغة ، فعلى القارىء استبدال PRINT بالجملتين التاليتين :

```
   WRITE(6, 31) N, K
31 FORMAT(2I15)
```

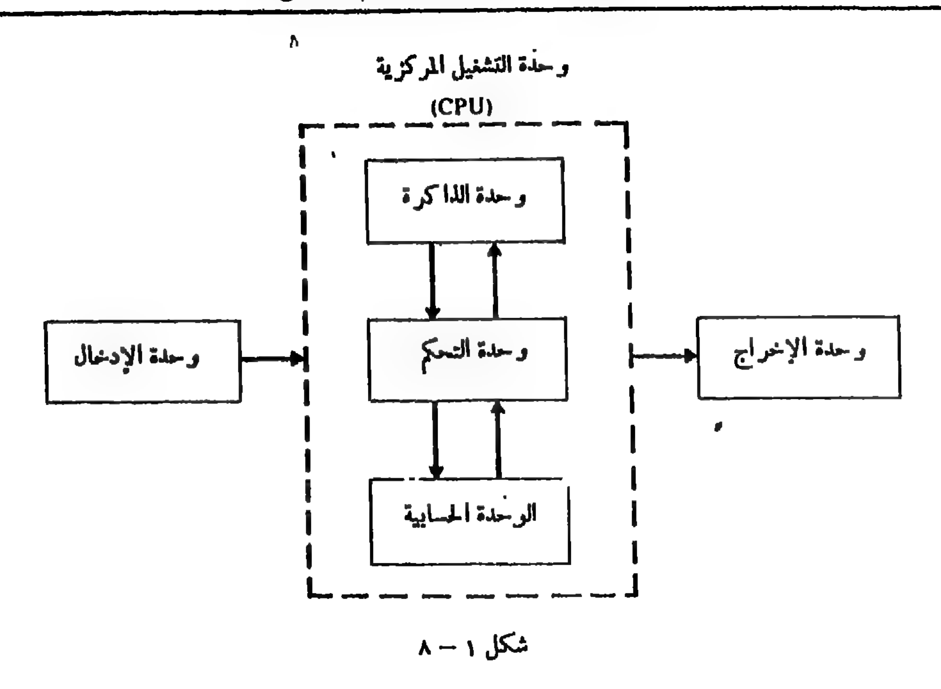
١ – ٧ نظرة عامة على الحاسبات واللغات

يعطى هذا القسم نظرة عامة على الحاسبات وعلى لغات الحاسب . ورغم أن القارىء يمكن أن يمر على هذا القسم مروراً سريعا فى القراءة الأولى للكتاب ، إلا أنه توجد بعض مصطلحات وتفاصيل معينة ومطلوبة حتى يمكن تفهم المواضيع اللاحقة ، وعلى سبيل المثال ، حمل منفذة وغير منفذة . ولقد ضمنا هذا القسم كل الضروريات لكى يكون الكتاب كاملا ، ويستطيع القارىء استخدام هذا القسم كمرجع عند الضرورة .

(أ) وحدات الحاسب

يتكون الحاسب من خمسة أجزاء أساسية : وحدة الإدخال ووحدة الإخراج ووحدة الذاكرة ووحدة التحكم والوحدة الحسابية . وهذه الوحدات موضحة فى شكل ١ – ٨ .

تتكون وحدات الإدخال والإخراج من أجهزة تسمح للحاسب باستقبال أو عرض المعلومات . وتكون فى صورة وحدات لقراءة البطاقات أو شرائط ممغنطة أو وحدات قراءة الشرائط الورقية أو وحدات الطباعة أو وحدات CRT (أنبوبة أشعة المهبط) طرفية أو وحدات التحكم الطرفية الكاتبة . بعض هذه الوحدات يمكن استخدامها للإدخال والإخراج معا .



شكل ١ — ٨

وتكون الوحدات الثلاث (وحدة الذاكرة ووحدة التحكم والوحدة الحسابية) مايسمى بوحدة التشغيل المركزية (CPU) . والغرض من هذه الوحدات الثلاث هو :

١ — وحدة الذاكرة : هي الجزء الذي تخزن فيه كل البيانات والنتائج . وتسمى الحاسبات الحالية بحاسبات البرامج المخزنة إذ أن البرامج نفسها تخزن أيضاً في الذاكرة . وتتكون وحدة الذاكرة من خلايا كثيرة جداً ، كل منها قادر على تخزين وحدة من المعلومات (كلمة) . وتسمى أيضاً هذه الخلايا ذات الطبيعة الكهرومغناطيسية ، بمراكز التخزين (متغيرات) . وترقم الخلايا بالتسلسل للتفرقة بينها ويشار إلى أرقامها أحياناً بالعنوان . أما بالنسبة للمستفيد ، فيعطى هذه العناوين في شكل رموز (أسماء) .

٢ — وحـده التحكم : تتحكم في أنشطة الأجزاء المختلفة للحاسب. فهي ترسل الأوامر وإشارات التحكم وتحدد أسبقية التعليمات المختلفة.

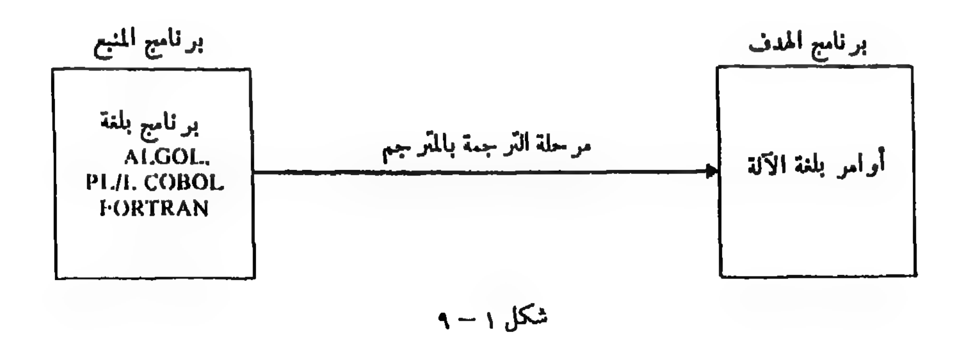
٣ — الوحـدة الحسابية : تتكون من دوائر كهربائية وتقوم بالعمليات الحسابية والمنطقية المختلفة .

(ب) الترتيب الهرمى للغـات البرمجـة

لكل حاسب لغة الآلة الخاصة به . ويجب أن تعطى التعليمات للحاسب بهذه اللغة حيث أنها اللغة الوحيدة التي يفهمها الحاسب . إلا أن هذه اللغة تختلف من حاسب لآخر ، وتعتمد على المكونات المادية للحاسب . وتعتمد البرامج المكتوبة بلغة الآلة على الآلة نفسها ولاتصلح إلا لها فقط .

تمثل تعليمات لغة الآلة عادة بالأعداد الثنائية ، أى سلسلة مكونة من آحـاد وأصفار (1's and 0's) ، والبرمجة بلغة الآلة معقدة وتفصيلية . وفى درجة أعلى من سلم الترتيب الهرمى للغات البرمجة توجد لغة التجميع . وفيها تستبدل سلاسل الآحاد والأصفار بالرموز بحيث تعطى الأوامر بأكواد رمزية تسمى نيموليك . وعلى ذلك نحتاج إلى برنامج مجمع لترجمة لغة التجميع إلى لغة الآلة . وبما أن تركيب لغة التجميع مشابه جداً للغة الآلة . فهذا يتطلب من مخططى البرامج أن يهتموا بالتفاصيل المتعددة مثل الفهرسة وأماكن التخزين بالإضافة إلى كتابة سلسلة من الأوامر المعقدة .

والآن نستطيع ، أن نكتب برامج الحاسب بلغة مقاربة للغة الإنجليزية مثل FORTRAN و COBOL و ALGOL و PI/I الخ . وتسمى هذه بلغات المترجم (رفيعة المستوى) وتكاد تكون مستقلة عن الآلة ، ويمكن بتعديلات بسيطة استخدام البرامج المكتوبة بهذه اللغات على الآلات المختلفة طالما وجد المترجم الخاص بها على هذه الآلة . والمترجم هو برنامج خاص يترجم تعليمات البرنامج المكتوب باللغات رفيعة المستوى إلى لغة الآلة . ويسمى البرنامج الأصلي ببرنامج المنبع وترجمته تسمى ببرنامج الهدف . (أنظر شكل ١ – ٩). من المهم أن نتذكر أن اللغات رفيعة المستوى تترجم أولا إلى أوامر بلغة الماكينة (الآلة) قبل أن ينفذها الحاسب فعلا .



شكل ١ – ٩

رغم أن اللغات رفيعة المستوى أقل كفاءة من لغات الآلة أو لغات التجميع ، إلا أنها تخلص مخطط البرامج من عبء الاحتفاظ بالتفاصيل الدقيقة ، مثل أماكن التخزين ، الخ ، بالإضافة إلى أنها أسهل في التعلم والاستخدام .

وكما ذكرنا سابقاً ، أن اللغة رفيعة المستوى التي ستتعلمها تسمى FORTRAN وسنستخدم أيضاً المصطلح «حاسب » بمعناه الواسع أي متضمناً المترجم ، ووحدة الطباعة والوحدات الإضافية الأخرى .

(ح) الجمل المنفذة وغير المنفذة .

كما سبق وأن ذكرنا أنه سيتم أولا تناول برنامج الفورتران بواسطة مترجم فورتران وهو يترجم البرنامج إلى أوامر بلغة الآلة . فبعض جمل الفورتران يقصد بها إمداد المترجم بمعلومات فقط ، ولا ينتج عنها أي أوامر بلغة الآلة . وتسمى هذه الجمل جملا غير منفذة . بمعنى آخر « فالجمل القابلة التنفيذ » هي تلك التي ينتج عنها نوع ما من الأوامر بلغة الآلة .

فجملة STOP جملة قابلة التنفيذ ، والمقابل لها في لغة (الآلة) أن يتوقف الحاسب عن تنفيذ أي أمر من أوامر بلغة الآلة التالية في البرنامج . ومن ناحية أخرى فجملة END جملة غير منفذة . فهي تخبر الحاسب أن هذه هي نهاية البرنامج ، وأنه لا توجد هناك جمل فورتران أخرى مطلوب ترجمتها إلى أوامر بلغة الآلة . وحيث أن برنامج الفورتران يترجم بالكامل قبل تنفيذ أي أمر بلغة الآلة فهذا يفسر السبب في أن جملة END هي دائماً آخر جملة في برنامج الفورتران .

هناك جمل فورتران أخرى كثيرة غير قابلة التنفيذ سنذكرها في وقتها كلما تقدمنا في النص .

مسائل محلولة

١ – ١ حدد خرج البرنامج التالى :

```
C     FIRST SOLVED PROGRAM
      J = 1
      K = 3
      L = 2*J + K
      J = 3*J + 2*L
      K = K + 2
      L = J + K + L
      PRINT, J, K, L
      STOP
      END
```

تنفيـذ البـرنـامج

السطر الأول : يتخطى الحاسب هذه الجملة حيث أنها تعليق .

السطر الثانى : يخزن الحاسب 1 فى مكان التخزين المسمى J .

J ← 1

السطر الثالث : يخزن الحاسب 3 فى مكان التخزين المسمى K

K ← 3

السطر الرابع : نأمر الحاسب بحساب قيمة 2*J + K وبعد ذلك تخزن النتيجة فى مكان التخزين المسمى L . وعلى ذلك :

$2*J + K = 2 \cdot 1 + 3 = 5$

ثم يخزن الحاسب 5 فى L

L ← 5

السطر الخامس : نأمر الحاسب بإيجاد قيمة 3*J + 2*L باستعمال القيم الحالية لـ J و L وبعد ذلك يستبدل القيمة الحالية لـ J بالناتج وعلى ذلك .

$3*J + 2*L = 3 \cdot 1 + 2 \cdot 5 = 13$

ثم يمسح الحاسب القيمة الحالية لـ J ويستبدلها بـ 13 ونبين هذا بالآتى :

J ← ~~1~~ 13

السطر السادس : نأمر الحاسب بزيادة القيمة الحالية لـ K بمقدار 2 .

K ← ~~3~~ 5

السطر السابع : يجمع الحاسب القيم الحالية لـ J و K و L .

$J + K + L = 13 + 5 + 5 = 23$

وبعد ذلك ، يمسح الحاسب القيمة الحالية لـ L ويستبدلها بـ 23

L ← ~~5~~ 23

السطر الثامن : يطبع الحاسب القيم الحالية لـ J و K و L ، أى :

13,	5,	23

السطر التاسع : يأمر الحاسب بالتوقف STOP

الخرج

يتكون الخرج من الأعداد 13 و 5 و 23 التى تم طباعتها على سطر واحد حيث أن جملة PRINT استعملت مرة واحدة فقط .

١ – ٧ حدد خرج البرنامج التالى .

```
C       SUM OF SQUARES
        K = 1
        JSUM = 0
     31 JSUM = JSUM + K**2
        K = K + 3
        IF(K.LT.10) GO TO 31
        PRINT, JSUM
        STOP
        END
```

حدد الخرج إذا ما استبدل السطر السادس من البرنامج بما يلى :

```
        IF(K.LT.20) GO TO 31
```

تنفيـذ البـرنـامج

الخـطوة :

١ – السطر الأول : يتخطى الحاسب هذه الجملة حيث أنها تعليق .

٢ – السطر الثـانى : K ← 1

٣ – السطر الثالث : JSUM ← 0

٤ – السطر الرابع : يحسب الحاسب قيمة JSUM + K**2 (حيث K**2 فى الفورتران تداول K^2) باستعمال القيم الحالية لـ JSUM و K :

$$JSUM + K{*}{*}2 = 0 + 1^2 = 1$$

بعد ذلك يمسح الحاسب الصفر الموجود فى SUM ويستبدله بالقيمة 1 .

JSUM ← ~~0~~ 1

٥ – السطر الخامس : يزيد الحاسب قيمة K بمقدار 3 .

K ← ~~1~~ 4

٦ – السطر السادس : يسأل السؤال :

هل K < 10 ؟

وحيث أن K = 4 ، فالإجابة نعم ، أى أن ، الجملة بين القوسين صحيحة . ومن ثم يذهب الحاسب إلى الجملة التى تحمل رقم 31 ، أى السطر الرابع فى هذا البرنامج .

٧ – السطر الرابع : يأمر الحاسب بزيادةقيمة JSUM بمقدار K^2 . وحيث أن K=4 يجمع الحاسب 16 على القيمة الحالية لـJSUM

JSUM ← ~~0~~ ~~1~~ 17

٨ – السطر الخامس : K ← ~~1~~ ~~4~~ 7

٩ – السطر السادس : يذهب الحاسب إلى الجملة التى تحمل رقم 31 ، وهى السطر الرابع حيث أن "K < 10" صحيحة .

١٠ – السطر الرابع : يجمع الحاسب 49 على القيمة الحالية . لـ JSUM حيث أن K = 7 .

JSUM ← ~~0~~ ~~1~~ ~~17~~ 66

١١ – السطر الخامس : K ← ~~1~~ ~~4~~ ~~7~~ 10

١٢ – السطر السادس : بما أن K = 10 فالجملة

K < 10

غير صحيحة ومن ثم يذهب الحاسب إلى الجملة التالية فى البرنامج .

١٣ – السطر السابع : يطبع الحاسب القيمة الحالية لـ JSUM وهى

66

١٤ – السطر الثـامن : يأمر الحاسب بالتوقف STOP .

الخـــرج

الخرج هو العدد الصحيح 66 فقط وهى القيمة الأخيرة لـ JSUM لاحظ أن هذه القيمة النهائية لـ JSUM هى مجموع :

$$JSUM = 1^2 + 4^2 + 7^2$$

أى مجموع مربعات القيم الصحيحة فى السلسلة التى تبدأ بـ 1 وتزاد بخطوات مقدارها 3 وتنتهى قبل أن تصل إلى القيمة 10.

فى حالة ما إذا استبدل السطر السادس بما يلى :

```
IF(K.LT.20) GO TO 31
```

سيستمر المجموع إلى ماقبل 20 أى أن القيمة النهائية لـ JSUM ستكون

$$JSUM = 1^2 + 4^2 + 7^2 + 10^2 + 13^2 + 16^2 + 19^2 = 952$$

وفى هذه الحالة سيكون الخرج هو العدد الصحيح 952 .

١ – ٣ إشرح الغرض من كل برنامج من البرامج التالية :

(أ)

```
READ, R
AREA = 3.1416*R*R
PRINT, R
PRINT, AREA
STOP
END
```

(ب)

```
READ, A, B, C, D
PROD = A*B*C*D
PRINT, A, B, C, D
PRINT, PROD
STOP
END
```

(أ) يحسب هذا البرنامج مساحة (AREA) الدائرة ذات نصف القطر R المعلوم .

(ب) يحسب هذا البرنامج حاصل الضرب (PROD) لأربعة أعداد معطاة هي : A و B و C و D

مســـائل تكميلية

١ – ٤ أوجد الخرج للبرنامج التالى :

```
C       FIRST SUPPLEMENTARY PROGRAM
        I  = 3
        J  = 2
        K = 2*I + J
        I  = I + 3
        J  = I + 3*J - K
        K = 2*J + 3*K
        PRINT, I, J, K
        STOP
        END
```

١ – ٥ حدد الخرج للبرنامج التالى :

```
        K = 3
        JSUM = 0
   51   JSUM = JSUM + K
        K = K + 4
        IF(K.LT.25) GO TO 51
        PRINT, JSUM
        STOP
        END
```

١ – ٦ أوجد الخرج للبرنامج التالى :

```
C       THIRD SUPPLEMENTARY PROGRAM
        I = 1
        J = 2
   71   K = 2*I + 3*J
        I = J + K
        J = I + 2*J
        IF(K.LT.100) GO TO 71
        PRINT, I, J, K
        STOP
        END
```

١ – ٧ حدد الخرج للبرنامج التالى :

```
C       FOURTH SUPPLEMENTARY PROGRAM
        I = 1
        PRINT, I
        J = 2
        PRINT, J
     41 K = I + J
        PRINT, K
        I = J
        J = K
        IF(K.LT.99) GO TO 41
        STOP
        END
```

١ – ٨ أوجد الخرج للبرنامج التالى :

```
C       FIFTH SUPPLEMENTARY PROGRAM
        J = 1
        K = 3
     11 PRINT, J
     12 J = J + 1
        IF(J.LT.K) GO TO 11
        K = K + 3
        IF(J.LT.20) GO TO 12
        STOP
        END
```

اجابات للمسائل التكميلية المختارة

١ – ٤ تطبع الأعداد الصحيحة 6 و4 و 32 على سطر واحد .

١ – ٥ يطبع العدد الصحيح 78 .

١ – ٦ تطبع الأعداد الصحيحة 568 و 776 و 464 على سطر واحد .

١ – ٧ تطبع الأعداد الصحيحة : 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 على أسطر مختلفه .

١ – ٨ يطبع الحاسب الأرقام من 1 إلى 20 تاركاً الرقم3 ومضاعفاته على أسطر مختلفة ، أى: 20, 19, 17, 16, 14, 13, [illegible]

الفصل الثانى

جمـــل رياضـــية

٢ ـ ١ مقدمـــة

لقد وضحنا فى الفصل الأول كيف يقرأ الحاسب بعض برامج الفورتران البسيطة وكيفية تنفيذها ، والآن سندرس لغة الفورتران بصورة منهجية حتى نستطيع أن نكتب برامج لحل المسائل المختلفة .

فيما يلى قائمة بالحروف المستخدمة فى لغة الفورتران :

أرقام :	0	1	2	3	4	5	6	7	8	9			
حروف هجائية :	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
رموز :	+	−	*	/	.	,	'	=	$	(	)		

هناك أيضاً المسافة الخالية والتى تعرف كحرف . سنشير إلى المسافة الخالية أحياناً بالحرف "b" ويكتب b كدليل . تسمى الأرقام مع الحروف الأبجدية ، بالحروف الأبجدية الرقية . ولا توجد حروف صغيرة .

وأحياناً يتم التفرقة بين الرقم 0 (صفر) وبين الحرف O بكتابة شرطة على الحرف O بالصورة التالية Ø .

٢ ـ ٢ ثوابت عددية (الاعداد)

هناك نوعان متميزان من الأعداد فى الفورتران ، الأعداد الصحيحة والأعداد الحقيقية . وبصورة تقريبية ، فالاعداد الصحيحة هى تلك التى نستعملها فى العد ، والاعداد الحقيقية هى تلك التىنستعملها فى القياس . ويختلف الحساب والتمثيل (داخليا وخارجيا) لهذين النوعين من الأعداد فى الفورتران . وسنناقش التمثيل الخارجى الخاص بكل نوع أى الاشكال التى نكتبها ويطبعها الحاسب . بينما ستتم مناقشة التمثيل الداخلى أى الاشكال التى يستخدمها الحاسب لتخزين البيانات بداخل الذاكرة فى الملحق (أ) .

يسمى الرقم الصحيح أيضاً الثابت ذو النقطة الثابتة وهو أى عدد صحيح له أو ليس له إشارة جبرية وبدون علامة عشرية أو أى علامات وقف أخرى . وسنعتبر الأرقام بدونالإشارة الجبرية أرقاماً موجبة . وأقصى طول لأى رقم صحيح لا يتعدى عادة 9 خانات ، غير إنها تختلف من حاسب إلى آخر . وننصح مخططى البرامج بفحص إمكاناتهم الحسابية المحلية لمعرفة تلك الصفات التى تعتمد على الآلة .

مثال ٢ ـ ١

الاعداد التـــالية مقبولة كثوابت صحيحة .

15 −8 0 12547 −5286

بينما الأعداد التـــالية غير مقبولة كثوابت صحيحة :

21.0 (يحتوى على علامة عشرية)

— 248. (يحتوى على علامة عشرية)

12,357 (يحتوى على فاصلة)

125000000000 (كبير جداً)

يمكن أن نكتب الثوابت الحقيقية فى الفورتران ، والتى تسمى أيضاً بثوابت النقطة الطليقة فى شكلين مختلفين : الشكل العشرى والشكل الأسى . فى كلا الشكلين توجد سلسلة محددة من الخانات مع العلامة العشرية . وفى الواقع ، فإن العلامة العشرية هى التى تميز الثابت الحقيقى عن الثابت الصحيح .

الشكل العشرى : يكتب الثابت الحقيقى فى الشكل العشرى كسلسلة محددة من الخانات سواء بإشارة جبرية أو بدونها ، مع علامة عشرية . ولا يمكن أن توجد أى علامة وقف أخرى . من الممكن كتابة الرقم الحقيقى بأى عدد من الخانات المعنوية ، إلا أن ، عدد الخانات المعنوية التى يحتفظ بها الحاسب تتوقف على طول كلمة الحاسب . وبصورة عامة ، لا يحتفظ الحاسب بأكثر من سبعة أو ثمانية أرقام معنوية . مرة أخرى ، يجب على القراء مراجعة إمكاناتهم الحسابية لهذه الخصائص التابعة للآلة .

مثال ٢ — ٢

الأعداد التـــالية مقبولة كثوابت حقيقية :

28.3   −236.0   24.   +.0187   −234.182

بينما الأعداد التـــالية غير مقبولة كثوابت حقيقية :

4,356.2 (يحتوى على فاصلة) .

— 36 (لا توجد علامة عشرية) .

الشكل الأسى : تتكون الثوابت الحقيقية عندما تكتب فى الشكل الأسى (E - Form) من جزءين . الجزء الأول هو ثابت حقيقى فى الشكل العشرى (أى سلسلة محددة من الخانات سواء بإشارة جبرية أو بدونها مع علامة عشرية) . ويبدأ الجزء الثانى بالحرف E يتبعه ثابت صحيح سواء بإشارة جبرية أو بدونها ومكون من خانتين على الأكثر . ويفسر الجزء الثانى على أنه الأس للأساس 10 . على سبيل المثـــال :

23.1E−4

هو ثابت حقيقى فى الشكل الأسى (E-form) ويمثل 23.1×10^{-4} وفيما يلى أمثلة مقبولة لثوابت حقيقية فى الشكل الأسى (E-form)

24.123E04	$24.123 \times 10^4 = 241230.0$
−5.36E−3	$-5.36 \times 10^{-3} = -0.00536$
2.6E+2	$2.6 \times 10^2 = 260.0$

لا توجد أى قيود بخصوص مكان وضع العلامة العشرية عندما تكتب أو تدخل عدداً حقيقيا فى الشكل الأسى (E-form) . ومن ثم ، فيمكن كتابة أى ثابت حقيقى فى الشكل الأسى (E-form) بأكثر من طريقة ، على سبيل المثال :

23.1E−4   2.31E−3   0.0000231E2

كلها ن نفس الرقم ويقبلها الحاسب .

من الممكن دائماً أن يمثل الرقم الحقيقى فى الشكل الأسى المعيارى أى أن الرقم السابق لحرف E (المسمى بالجزء الحقيقى) يقع بين :

0.1 و 1.0 (أو 0.1— و 1.0—)

وفى واقع الأمر سوف يطبع الحاسب دائماً (أى الخرج) الثابت الحقيقى فى الشكل الأسى (E-form) بهذه الطريقة .

وكما ذكرنا سابقا . فإن عدد الخانات المعنوية التى يحتفظ بها الحاسب تتوقف على الآلة . بالإضافة إلى ذلك ، فإن الحد الأقصى المقبول للأس (عادة ما يحدد بخانتين على الأكثر) وهو يتوقف على الآلة أيضاً .

ملاحظة : يختلف تصور الثابت الحقيقى فى الفورتران عنه فى الرياضيات . فى الواقع فإن كل الأرقام الحقيقية فى الرياضيات تقرب بأعداد منطقية فى الحاسبات .

رغم أن 24.0 و 2.4E1 مختلفان فى المظهر (الخارجى) إلا أنهما متطابقان فى التمثيل الداخلى حيث أن كليهما اعداد حقيقية (انظر ملحق أ) . ومن ناحية أخرى 24 و 24.0 لها تمثيل داخلى مختلف ويعاملان بصورة مختلفة فى الفورتران فأحدهما يدل على ثابت صحيح والآخر يدل على ثابت حقيقى (كلاهما يمثل نفس العدد رياضياً) .

لم نناقش حتى الآن ، كيف نأمر الحاسب أن يقرأ أو أن يطبع عدداً صحيحا و/أو عدداً حقيقياً . ستناقش هذه الأمور فى الفصل الثالث .

٢ — ٣ اسماء المتغيرات (اسماء اماكن التخزين)

يستخدم دائماً المصطلح « متغير » فى علم الحاسب ليعنى عنصر ذاكرة . والخصائص الأساسية التى يجب أن نتذكرها عن عناصر الذاكرة هى : القراءة الهدامة إلى الذاكرة والقراءة غير الهدامة من الذاكرة . أى أنه إذا خزنت معلومات فى خلية ذاكرة فسوف تتبدد المحتويات الأصلية (السابقة) فى خلية الذاكرة ولكن إذا نقلت (نسخت) معلومات من الخلية فستبقى المحتويات (الأصلية) فى الخلية كما هى بدون تغيير .

إذا استعمل مكان ذاكرة (متغير) لتخزين ثوابت صحيحة ، سمى متغيرا صحيحا أما إذا استعمل لتخزين ثوابت حقيقية سمى متغيرا حقيقيا .

تعطى الاسماء لخلايا الذاكرة ، لسهولة الرجوع إليها . وعموماً يرمز لأسم المتغير بواسطة عدد من الحروف الأبجدية تتراوح ما بين حرف إلى ستة حروف ، ويجب أن يكون الحرف الأول حرفاً أبجديا ، وعلى ذلك فالأسماء IAM, A2X3, BAD, DEAN هى متغيرات مقبولة أما الأسماء التالية فهى أسماء غير مقبولة .

2XY لأنها لا تبدأ بحرف أبجدى .

A*B6 لأن* ليست حرفاً أبجدياً ورقميا .

INTEREST تحتوى على أكثر من ستة حروف .

(تقبل بعض الحاسبات أسماء للمتغيرات أكثر من ستة حروف ، ولكن بالنسبة لمعظم الحاسبات فالحد الأقصى هو ستة حروف .)

يستعمل الفورتران أيضاً القاعدة الضمنية الآتية لأسماء المتغيرات : إذا كان أول حرف هو أحد الحروف الستة التالية I, J, K, L, M, N فهو اسم لمتغير صحيح ، أما إذا كان الحرف الأول أى حرف من الحروف الأخرى ، أى أحد الحروف التالية A, ..., H, O, P, ..., Z فهو اسم متغير حقيقى (يمكن إبطال هذه القاعدة بجملة اعلانية النوع التى ستناقش فى القسم القادم) .

مثال ٢ — ٣

المتغيرات التالية مقبولة كمتغيرات صحيحة :

MONEY NEXT J4X2 IDIOT

بينما المتغيرات التاليه مقبولة كمتغيرات حقيقية :

ANSWER X21 BUM ENEMY

ملاحظة : كلمات معينة مثل READ و WRITE و PRINT و PAUSE و STOP و END الخ ، هي كلمات دالة في لغة الفورتران ولذلك لا تقبل كأسماء متغيرات في معظم الحاسبات .

٢ — ٤ جمل النوع — صحيح وحقيقي

في بعض الأحيان يكون غير مرغوب أن نتبع القاعدة السابقة لكتابة المتغيرات . فعلى سبيل المثال قد ترغب في استخدام (الرمز النيموني — مساعد الذاكرة) INT ليرمز إلى متغير حقيقي لسعر الفائدة . فلابطال القاعدة الضمنية يمدنا الفورتران بعدة جمل صريحة للاعلان عن النوع . فجملة الفورتران المبتدئة بكلمة REAL متبوعة بقائمة من المتغيرات والتي تفصل بين كل منها فصلة (،) تعلن أن هذه المتغيرات REAL (أي متغيرات حقيقية) . وعلى ذلك فإن

REAL INT

تعلن الحاسب بأن INT ترمز دائماً إلى متغير حقيقي طوال البرنامج ، وبطريقة مشابهة فإن جملة الفورتران المبتدئة بكلمة INTEGER تتبعها قائمة من أسماء المتغيرات تفصل بين كل منها فصلة (و) ، تعلن أن هذه المتغيرات INTEGER (أي متغيرات صحيحة) .

تمد جمل النوع المترجم بمعلومات ، ومن ثم فهي جمل غير منفذة . ويجب أن تظهر في بداية البرنامج . وسيتم تقديم جمل نوع أخرى غير INTEGER و REAL في فصول قادمة .

مثال ٢ — ٤

افترض أن برنامجا يبدأ بالجمل التالية :

```
INTEGER A, B, SUM, TIME
REAL NUMBER, MIN, MAX
```

وعلى ذلك ... نامج ستشير A وB و SUM و TIME إلى متغيرات صحيحة أما NUMBER و MIN و MAX فستشير إلى متغيرات حقيقية .

يشعر بعض محللي البرامج بأن ذكر كل المتغيرات في جمل نوع بغض النظر عن نوعية الحرف الأول من هذه المتغيرات ، تمرين يستحق الاهتمام فعلى سبيل المثال ، يمكننا كتابة :

```
REAL NUMBER, MIN, MAX, RATE
```

رغم أن RATE متغير حقيقي إن لم يكتب صراحة في جملة نوع .

٢ — ٥ عمليات حسابية — حسابات صحيحة وحقيقية

يبين الجدول ٢ — ١ الرموز الخمس للعمليات الحسابية الأساسية في الفورتران ونحن نؤكد أن التعبيرات الجبرية :

$a \cdot b \qquad \frac{a}{b} \qquad a^b$

تكتب : A*B, A/B, A**B على الترتيب .

جـدول ٢ – ١

العملية	الرمز
جمع	+
طـرح	−
ضرب	*
قسمة	/
الرفع للأس	**

حيث أن هناك نوعين من الأرقام ، صحيحة وحقيقية فهناك كذلك نوعان من الحسابات حسابات حقيقية وحسابات صحيحة ويتم إنجاز هذين النوعين من الحسابات بطرق مختلفة ولذا فن المهم جداً أن يكون الفرق بينهما مفهوماً فهما واضحا .

(أ) حسابات صحيحة

إذا كانت المعاملات أعداداً صحيحة ، تنفذ الحسابات الصحيحة لتنتج عددا صحيحا . على سبيل المثال ، تستخدم الحسابات الصحيحة لإيجـاد قيمة : 5 + 3 ,5*3 و 5 − 3

فيكون الناتج 8 و 15 و 2 على الترتيب . وفى القسمة الصحيحة فى الفورتران عند إيجاد قيمة I/J حيث كل من I و J عدد صحيح يكون الناتج أيضاً صحيحاً وهو الجزء الصحيح من خارج القسمة . وعلى ذلك يبتر الجزء الكسرى من خارج القسمة فى القسمة الصحيحة . على سبيل المثال :

5/4 تعطى 1 و 4/5 تعطى 0 و 7/2— تعطى 3-.

وهكذا فالقسمة الصحيحة فى الفورتران تختلف تماماً عن القسمة العادية ، ومع ذلك يمكن استخدامها فى صالحنا . كما هو موضح فى المثال ٢ – ٥ .

(ب) حسابات حقيقية

إذا كانت المعاملات حقيقية ، تنفذ الحسابات الحقيقية لتنتج قيما حقيقية . وعلى ذلك فالحسابات الحقيقية تستخدم لإيجاد :

5. + 3 ، 5.*3. و 5. —3.

لتعطينا القيم الحقيقية .8 ، .15 ، .2 على الترتيب . والقسمة بالحسابات الحقيقية فى الفورتران مشابهة للقسمة العادية أى أن :

5./4 تعطى 1.25 و 4./5. تعطى 0.8 ، 7./2.— تعطى 3.5-.

أهم جزء فى المناقشة المذكورة اعلاه هو أن نوع المعامل يحدد نوع الحساب (أو نمط التنفيذ) ويكون الناتج محصلة من نفس النوع . ونلاحظ أن الرفع للأس يعامل بطريقة مختلفة . فالقيم الحقيقية الموجبة فقط هى القيم التى يمكن أن ترفع إلى قوى حقيقية بينما أى قيمة سواء صحيحة أو حقيقية يمكن أن ترفع إلى قوى صحيحة . وسنناقش ذلك فى القسم ٢ – ٨ .

مثال ٢ – ٥

مع فرض أن الاستخدام التقليدى للأقواس فى التعبيرات الحسابية مقبول أيضاً فى الفورتران . ناقش الفروق بين :

(أ) (N*2)/2 و (ب) (N/2)*2

بين أيضاً كيف يمكن أن نستخدم (ب) لتحديد ما إذا كانت N زوجية أو فردية .

التعبيرات الجبرية التى تمثلها هى :

$$\frac{n \cdot 2}{2} \quad و \quad \frac{n}{2} \cdot 2$$

على الترتيب : وهما متطابقان رياضيا . ولكنهما ، بالحساب الصحيح فى الفورتران غير متكافئين . وبالتحديد ، لدينا الآتى :

(أ) هنا نحسب قيمة N*2 أولا لتعطى عدداً زوجياً صحيحا ، ومن ثم (N*2)/2 دائماً ستأخذ القيمة N .

(ب) هنا نحسب قيمة N/2 أولا لتعطى الجزء الصحيح من خارج القسمة وعلى ذلك إذا كانت N زوجية ، فتعطى (N/2)*2 الناتج N ولكن إذا كانت N فردية فتعطى (N/2)*2 الناتج N ± 1 . على سبيل المثال إذا كانت N = 8 فإن :

(N/2)*2 = 4*2 = 8

وإذا كانت N = 11 فإن :

(N/2)*2 = 5*2 = 10

وإذا كانت N = 9 فإن :

(N/2)*2 = 4*2 = 8

٢ — ٦ التعبيرات الحسابية

يجب أن نفهم كيف يقوم الحاسب بإيجاد قيمة التعبيرات الحسابية حتى نستطيع كتابتها بطريقة صحيحة .

يوجد الحاسب قيمة التعبيرات الخالية من الأقواس (أى ، التعبير أو التعبير الفرعى بدون أقواس) ، باستعمال جدول الأسبقية التقليدى التالى :

أسبقية أولى : الأس (**)

أسبقية ثانية : الضرب (*) والقسمة (/)

أسبقية ثالثة : جمع (+) وطرح (−)

(ستعامل الزائد الأحادى والناقص على نفس المستوى مثل الجمع الثنائى والطرح أى 2.2**2 — تعنى (2.2 ** 2)— والتى تنتج 4.84) .

حيث أن هناك ثلاثة مستويات للأسبقية ، فتصور أن الحاسب يمسح (أو يمر على) التعبيرات الرياضية الخالية من الأقواس من اليسار إلى اليمين ثلاث مرات . المرة الأولى للبحث عن الأس ، والمرة الثانيةللبحث عن الضرب والقسمة ، وأخيراً ، للبحث عن الجمع والطرح . حيث أن الحاسب يمر من اليسار إلى اليمين ، فإذا كانت عمليتان لهما نفس الأسبقية ، فتنفذ أولا تلك التى على أقصى اليسار .

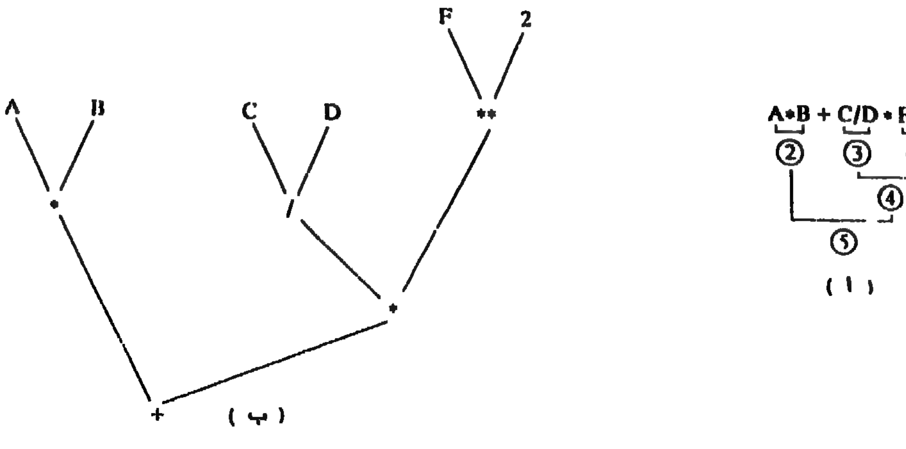
مثال ٢ — ٦

إدرس التعبير

A*B + C/D*F**2

يوضح شكل ٢-١ (أ) ترتيب التنفيذ . ويمكن لهذا الترتيب أن يكتب بصورة مكافئة كشجرة مقلوبة والمعاملات الحسابية كعقد (nodes) كما هو موضح فى شكل ٢ - ١ (ب) . لاحظ أن التعبير يمثل ما يلى :

$$a \cdot b + \frac{c}{d} \cdot f^2$$



شكل ٢ - ١

يسمح الفورتران أيضاً باستخدام الأقواس بالطريقة التقليدية . بمعنى أن الأقواس لها أسبقية أعلى من كل العمليات الحسابية ، وتعامل الأقواس الداخلية أولا .

مثال ٢ - ٧

(أ) يمكن ايجاد قيمة التعبير (X + Y/(Z * A + B**2 كالتـالى :

X + Y/(Z*A + B**2)

② ①

③

④

⑤

وهكذا فالتعبير الجبرى هو :

$$x + \frac{y}{z \cdot a + b^2}$$

(ب) المعادلة الجبرية :

$$x = [(a + b)^2 + (3c)^3]^{a/b}$$

يمكن أن تكتب بالفورتران كالتـالى :

X = ((A + B)**2 + (3.0*C)**3)**(A/B)

وننصح بشدة باستخدام أقواس إضافية كلما كان هذا الاستعمال مؤكدا للصواب . علاوة على ذلك ، إذا احتوى أى تعبير رياضى على حدود طويلة ومعقدة ، فنحبذ دائماً تقسيمها إلى عدة جمل أصغر وأقل عرضة للخطأ . على سبيل المثال التعبير الجبرى :

$$y = \left[\frac{ab}{c+d} - \frac{g}{5(h+x)}\right]^{1/r}$$

يجب أن تمثل كما يلى :

```
T1 = A*B/(C + D)
T2 = G/(5.*(H + X))
 Y = (T1 - T2)**(1./R)
```

ملاحظة : لا يسمح الفورتران بظهور عمليتين حسابيتين متجاورتين . وعلى هذا A* — B يجب أن تكتب A* (—B)

مثال ٢ — ٨

فيما يلى بعض تعبيرات رياضية وتعبيرات الفورتران المقابلة لكل :

	فورتران	رياضيا
(أ)	$a + \frac{b}{c} + d$	A + B/C + D
(ب)	$\frac{a+b}{c+d}$	(A + B)/(C + D)
(جـ)	$a^3 - b^3$	A**3 - B**3
(د)	$\frac{ab}{c - d^2}$	A*B/(C - D**2)
(هـ)	$\frac{ab}{c} - d^2$	A*B/C - D**2
(و)	$\frac{a}{bc} - d^2$	A/(B*C) - D**2
(ز)	$1 + \frac{a}{b + \frac{1}{c}}$	1. + A/(B + 1./C)

لاحظ ضرورة وجود الأقواس فى (و) ولكنها غير ضرورية فى (هـ) .

٢ — ٧ العمليات الحسابية ذات النمط المختلط

وكما ناقشنا فى القسم ٢ — ٥ ، حينما يحسب الحاسب قيمة تعبير رياضى فإن نوع حدود التعبير تحدد نوع الحساب المستخدم (نمط العملية) . وعلى ذلك ، يجب أن تكون كل المتغيرات والثوابت فى أى تعبير رياضى بالفورتران من نفس النوع ، أى يجب أن تكون جميعها إما حقيقية أو صحيحة . وبالتالى إذا كانت :

A + 3

هو تعبير أحد حدوده A متغير حقيقى والآخر صحيح ، فيمكن أن نتساءل :

(١) عما إذا كان مسموح بمثل هذا النمط المختلط فى التعبير .

(٢) ومتى إذا كان مسموحا به فما هى نوعية الحسابات التى سوف تنفذ .

لا تسمح بعض مترجمات (Compilers) الفورتران بالتعبيرات الرياضية ذات النمط المختلط . في مثل هذه الحالة ، يجب أن تحتاط لتجنب الخلط بين نوعين في تعبير واحد . وبذلك فالتعبيرات

A + 3*B	يجب أن تكتب	A + 3.*B
I + 4.0*J	يجب أن تكتب	I + 4*J
B**2 – 4*A*C	يجب أن تكتب	B**2 – 4.*A*C

لا يعتبر رفع رقم حقيقي إلى قوة صحيحة نمط عمليات مختلط (أنظر المناقشة في نهاية قسم ٢ – ٨) ، وعلى هذا B**2 يسمح بها ولا يجب أن تتغير إلى B**2.0 (لمعرفة الفرق بين B**2 و B**2.0 أنظر قسم ٢ – ٨ وأيضا المسألة ٢ - ١٧) .

أغلب الحاسبات الكبرى لديها مترجمات تقبل تعبيرات النمط المختلط . في مثل هذه الحالة ، إذا كانت حدود إحدى العمليات من نوعين مختلفين أي إحداهما صحيح والآخر حقيقي ، فيحول الصحيح أولا إلى حقيقي ثم تجرى الحسابات الحقيقية لإعطاء قيمة حقيقية. وقد تبدو هذه وسيلة مريحة حيث يمكن ظهور كلا من المتغيرات والثوابت الصحيحة والمتغيرات والثوابت الحقيقية بعد ذلك في تعبير رياضي واحد ، ولكن يجب على مخططي البرامج أن يحترسوا من عواقب ظهور كلا النوعين من المتغيرات والثوابت في تعبير واحد .

مثال ٢ – ٩

(أ) أدرس التعبير الرياضي ، بفرض أن حساب النمط المختلط مسموح به

(11./2) + 4.3

وحيث أن .11 و 2 نوعان مختلفان فيحول العدد الصحيح 2 إلى .2 وتجرى الحسابات الحقيقية وينتج عنها القيمة الحقيقية 5.5 وبجمع الأرقام الحقيقية 5.5 و 4.3 فإننا نحصل على النتيجة 9.8 .

(ب) أدرس التعبير الرياضي :

(11/2) + 4.3

نحسب قيمة الحد 11/2 أولا . وحيث أن 11 و 2 أعداد صحيحة ، من ثم تحسب قيمة 11/2 بواسطة الحساب الصحيح فتكون النتيجة 5 (وليست 5.5) . وحيث أن حدود الجمع من أنواع مختلفة ، فيحول العدد الصحيح 5 إلى عدد حقيقي ويضاف إلى الثابت الحقيقي 4.3 باستعمال الحساب الحقيقي مما يعطى العدد الحقيقي 9.3 .

يجب أن يدرك مخططو البرامج أن وسيلة الراحة هذه لاتأتي « بدون مقابل » . كلا ، ففي كل مرة تحول قيمة صحيحة إلى حقيقية ستهلك فيها وقت الآلة . وبذلك فالسماح بالتعبيرات الرياضية ذات النمط المختلط سيزيد من وقت التنفيذ ، وسيعرض البرنامج إلى خطر التقليل من كفاءة التنفيذ . بالإضافة إلى ذلك ، فالبرامج التي تكتب بالنمط المختلط أقل تداولا . أي قد لاتستطيع تشغيلها على حاسبات مختلفة . وأخيراً فاستخدام النمط المختلط بدون تفكير قد ينتج قيما غير مقصودة كما هو موضح ضمنياً في مثال ٢ – ٩ .

٢ – ٨ الدوال الرياضية المبيتة (المبنية ذاتيا)

للغة الفورتران مكتبة للدوال الرياضية التي يمكن أن يستخدمها مخطط البرامج . بعض هذه الدوال الأكثر شيوعاً معروضة في الجدول ٢ – ٢ . وهناك قائمة أخرى أكثر شمولا معطاة في الملحق 1

جدول ٢ - ٢

الدالة	الرمز الرياضي	في الفورتران
الجذر التربيعي للمتغير x	$\sqrt{x}$	SQRT(X)
القيمة المطلقة للمتغير x	$\|x\|$	ABS(X)
الرفع إلى أس x	e^x	EXP(X)
جيب الزاوية x	$\sin x$	SIN(X)
جيب تمام الزاوية x	$\cos x$	COS(X)
لوغاريتم x	$\log_{10} x$	ALOG10(X)
اللوغاريتم الطبيعي لـ x	$\log_e x$	ALOG(X)
تحويل العدد الصحيح I إلى عدد حقيقي		FLOAT(I)
بتر العدد الحقيقي x إلى عدد صحيح .		IFIX(X)

رغم أننا سنناقش دوال البرامج الفرعية بالتفصيل في الفصل السابع إلا إننا سوف نشرح هنا بعض الرموز والمصطلحات . فيرمز التعبير

SQRT(X)

للقيمة التي تخصصها دالة الجذر التربيعي للمتغير X . وتسمى X خلاصة الدالة ، ونطلق على SQRT اسم الدالة . لاحظ أن اسم كل دالة يتبعه قوسين يحيطون بخلاصة الدالة .

يستخدم جدول ٢ - ٢ كلا من X و I كخلاصات للدوال . تشير X إلى أن الخلاصة يجب أن تكون حقيقية ، كما تشير I إلى أن الخلاصة يجب أن تكون من النوع الصحيح . وفي الجانب الآخر يتحدد نوع قيمة الدالة بواسطة إسم الدالة تبعاً للقوانين العادية لأسماء لتسمية المتغيرات وعلى سبيل المثال :

ALGO (X)	تدل على قيمة حقيقية مناظرة للخلاصة الحقيقية X .
IFIX(X)	تدل على قيمة صحيحة مناظرة للخلاصة الحقيقية X .
FLOAT(I)	تدل على قيمة حقيقية مناظرة للخلاصة الصحيحة I .

وهكذا :

فالدوال FLOAT و IFIX ليست دوال رياضية عادية . تحول دالة FLOAT القيمة الصحيحة إلى القيمة الحقيقية المكافئة لها . فعلى سبيل المثال :

تعطى FLOAT(4) القيمة 4.0 وتعطى FLOAT(— 25) القيمة 25.0 — .

وتستخدم دالة FLOAT عادة لتجنب استخدام حساب النمط المختلط على سبيل المثال ; إذا كانت SUM تشير إلى مجموع عدد N من الأرقام ، حينئذ يمكن أن نكتب :

AVE = SUM/FLOAT(N)

وذلك للحصول على المتوسط AVE لعدد N من الأرقام . تحول دالة IFIX قيمة حقيقية إلى قيمة صحيحة بواسطة البتر ، فعلى سبيل المثال ،

تعطى IFIX(2.9) القيمة 2 وتعطى IFIX(— 3.7) القيمة 3 —

وأننا لنتأكد أن دالة IFIX لاتقرب القيم الحقيقية إلى أقرب قيمة صحيحة .

يجب أن نأخذ في الاعتبار الملاحظات التالية :

١ — لانستلزم أن تكون خلاصات الدالة متغيرات مفردة . ولكن يمكن أن تكون تعبيرات حسابية من النوع المطلوب . فمثلا ترجمة $\sqrt{b^2 - 4ac}$ بالفورتران ستكون SQRT(B**2 - 4.*A*C)

٢ – يمكن أن تظهر الدالة كجزء من أي تعبير حسابي طالما نعطي الخلاصة المناسبة . فمثلا ترجمة :

$|a - b| + \frac{c}{\sin d}$ بالفورتران ستكون ABS(A – B) + C/SIN(D)

٣ – يمكن أن تستدعى خلاصة الدالة دالة أخرى ، أي يمكن أن يكون لدينا دوال لدوال ، فمثلا ترجمة :

$\sin\left|\frac{x - y}{x + y}\right|$ بالفورتران ستكون SIN(ABS((X – Y)/(X + Y))) .

٤ – لإيجاد قيمة التعبيرات الحسابية الخالية من الأقواس والتي تتضمن دوالا مكتبية سبقة (مبنية داخليا) يكون لحساب قيمة هذه الدوال أسبقية على القوانين التي تمت مناقشتها في قسم ٢ – ٦ . فمثلا ، التعبير :

```
ABS(A – B) + C/SIN(D)
   └──┘        └────┘
    ①            ②
    │       └────┘
    │         ③
    └─────────┘
         ④
```

سيتم حساب قيمته بالترتيب الموضح . لاحظ أن خلاصة الدالة ABS تعبير ، ومن ثم فعند حساب قيمة الدالة سوف نطبق قوانين الأسبقية مرة ثانية .

مثال ٢ – ١٠

فيما يلي تعبيرات رياضية وتعبيرات الفورتران المناظرة لها

	رياضيا	فورتران
(أ)	$\frac{\sin x}{\lvert y\rvert + \cos z}$	SIN(X)/(ABS(Y) + COS(Z))
(ب)	$\frac{e^{x+y}}{x + y}$	EXP(X + Y)/(X + Y)
(حـ)	$e^{\lvert x-y\rvert} + x$	EXP(ABS(X – Y)) + X
(د)	$a + \frac{b}{\lvert m - n\rvert}$	A + B/ABS(FLOAT(M – N))

خطأ شائع يقع فيه كثير من مخططي البرامج المبتدئين عند كتابة الثابت π . وحيث أن π ليس ثابتاً مبيناً ، وكذلك ليس حرفاً مقبولا في الفورتران ، فيجب استخدام قيمة مقربة لـ π . فمثلا ، تعبير الفورتران :

$\sin(2n\pi + x)$ سيكون SIN(2.*N*3.14 + X)

حيث تستخدم 3.14 بدلا من π . وبالمثل ، يمكن أن تخصص 3.14 للمتغير PI (أنظر قسم ٢ – ٩) وبعد ذلك نكتب :

SIN(2.*N*PI + X)

ننهي هذا القسم بالنقاط البسيطة الهامة التالية ، والمتعلقة بالرفع إلى أس حيث أنها تتضمن الدالة اللوغاريتمية .

١ – حين ترفع قيمة لقوة حقيقية كما في X**Y فإن قيمتها تحسب داخلياً كما يلي :

$e^{y \ln(x)}$

حيث ln هي دالة اللوغاريتم الطبيعي ، أى (EXP(Y*ALOG(X . (أنظر جدول ٢-٢) - وحيث أن خلاصات الدالة اللوغاريتمية يجب أن تكون حقيقية وموجبة ، لذا فقيمة X التى يمكن رفعها إلى قوى حقيقية يجب أن تكون حقيقية موجبة . ويجب أن يحتاط مخطط البرامج بصفة خاصة لهذا حيث أن خطأ رفع قيمة سالبة إلى قوة حقيقية يكتشف فقط أثناء وقت التنفيذ.

٢ - عندما يرفع رقم إلى قوة صحيحة كما فى X**I فإن ذلك ينفذ بالضرب . ومن ثم ، فيمكن رفع أى قيمة ، صحيحة أو حقيقية ، سالبة كانت أو موجبة ، إلى قوة صحيحة (أيضاً ، أنظر المسألة ٢ - ١٧) .

٣ - رياضياً ، a^{bc} تعنى $a^{(bc)}$ وهى تختلف عن $(a^b)^c$. ومع ذلك فإن A**B**C يمكن أن تأخذ الصورة (A**B)**C فى بعض المترجمات والصورة (A**(B**C بواسطة مترجمات أخرى . ومن ثم فنحن نشجع مخططى البرامج على استخدام الأقواس عند كتابة مثل هذه التعبيرات . وعموماً فن الحكمة استخدام الأقواس ، كلما كان هناك شك .

٢ - ٩ جملة التخصيص الحسابية

لكى ينفذ الحاسب تعبيراً رياضياً ، يجب أن تعطى كل المتغيرات التى تظهر فى التعبير قيما فى مكان ما بالبرنامج ولكن قبل استخدامها ، أى يجب أن تعرف مسبقا . تعطى معظم الحاسبات رسالة خطأ إذا لم يعرف أى متغير فى تعبير . كيف نعرف المتغير؟ أى كيف تخزن قيمة فى الذاكرة ؟ يعرف المتغير أما بقراءة قيمة من مجموعة بطاقات البيانات (أنظر الباب الثالث) أو بجملة تخصيص حسابية .

رمزياً ، نشير إلى تخزين (تخصيص) الثابت 1.5 فى مكان يسمى X كما يلى :

X ← 1.5

وتسمى هذه جملة تخصيص . فى الفورتران ← تمثل بعلامة = ، وعلى ذلك X ← 1.5 تكتب فى الفورتران كما يلى :

X = 1.5

من ناحية أ رى ، نفرض أن x تم تعريفها فعلا ، ونريد حساب قيمة x^2 ثم تخزين الناتج فى المكان Y أى $Y \leftarrow x^2$. فى الفورتران ، يمكن أن نكتب

Y = X**2

وعموماً ، تأخذ جملة التخصيص الحسابية فى الفورتران الشكل التالى :

تعبير حسابى (رياضى) = اسم متغير

وأننا لنؤكد أن الجانب الأيسر من علامة = يجب أن يكون إسم لمتغير مفرد .

من سوء الحظ أن الرمز = يستخدم لجمل التخصيص فى الفورتران ، حيث أن = تستخدم تقليدياً لتعنى التساوى . وأننا ننصح القراء بشدة أن تتذكر دائماً أن = تعنى « تخصيص » أو « إحلال » أو « نسخ » ، الخ . فى مناقشتنا سنستعمل أحياناً الرمز ← للتخصيص فضلا من بديله فى الفورتران = حيث نجد أن هذا الاستخدام يسهل عملية الفهم .

ملاحظة : أنه لغاية فى الأهمية أن نعرف أن فعلين منفصلين يحدثان فى جملة التخصيص الحسابية :

١ - إيجاد قيمة التعبير الحسابى على الجانب الأيمن من التساوى .

٢ - تخصيص نتيجة (١) لمكان التخزين على الجانب الأيسر من علامة التساوى . من المقبول أن يكون جانباً علامة = من أنواع مختلفة . حينما يكون جانباً علامه = من أنواع مختلفة ، فيتضمن التخصيص فى الخطوة الثانية تحويل النتيجة فى (١) إلى نفس نوع مكان التخزين الموضح على يسار علامة التساوى . وبالتحديد ، ستبتر القيمة الحقيقية قبل أن تخصص إلى مكان تخزين صحيح .

يختلط الأمر على كثير من مخططى البرامج المبتدئين حول التعبيرات ذات النمط المختلط وجمل التخصيص ذات الأنواع المختلفة على جانبى علامة = . ويشير النمط المختلط إلى تعبيرات حسابية حيث المعاملات من أنواع مختلفة . كما تشير جمل التخصيص إلى حركة تخزين قيمة فى الذاكرة ، وهى أمر فورتران ويمكن أن يشمل حساب قيمة تعبير حسابى . والتعبير الحسابى نفسه ليس أمراً . لاتقبل بعض المترجمات التعبيرات ذات النمط المختلط . ولكن تقبل كل المترجمات جمل التخصيص بأنواع مختلفة على جانبى علامة = .

مثال ٢ – ١١

(أ) أدرس الجملة :

INCOME = PRIN*(1. + XINT)

التعبير الحسابى على الجانب الأيمن من = من النوع الحقيقى . تحول نتيجة حسابات الجانب الأيمن أولا إلى صحيح (يبتر الجزء الكسرى منها) قبل تخزينها فى INCOME حيث INCOME متغير صحيح . وبالتحديد ، اذا كانت XINT = 0.0625 و PRIN = 100.00 فالنتيجة PRIN*(1. + XINT) هى 106.25 ولكن قبل أن (تنسخ) فى INCOME يبتر الجزء الكسرى 0.25 وتخزن القيمة الصحيحة 106 فى INCOME .

(ب) أدرس الجملة :

ASSET = INCOME + LOOT

يحسب هنا التعبير الحسابى فى النمط الصحيح ويحول بعد ذلك إلى النوع الحقيقى قبل أن يخزن فى المكان الحقيقى المسمى ASSET .

(جـ) أدرس الجملة :

SUM = SUM + X

حيث أن هناك حركتين منفصلتين فى جملة التخصيص SUM + X تنفذ أولا . ومن ثم تجمع القيم الحالية لـ SUM و X وتخزن النتيجة حينئذ فى SUM مرة أخرى . وحقيقة فإن الجملة تقول :

« القيمة الجديدة لـ SUM هى نتيجة جمع القيمة القديمة لـ SUM و X »

من الواضح ، أنه إذا ترجمت العلامة = على أنها « تساوى » فالجملة المعطاة SUM = SUM + X ستكون بلا معنى على الإطلاق .

٢ – ١٠ رياضيات الحاسب

سنكون مقتصرين الغاية مالم نذكر القارىء ببعض العواقب الوخيمة لرياضيات الحاسب .

وحيث أن الحاسب يحتفظ فقط بعدد معين من الأعداد المعنوية (بسبب الطول المحدود للكلمة) ، فكل الأرقام داخل الحاسب أرقام حقيقية . وعلى هذا ، فمعظم الأرقام ، بالتحديد كل الأرقام غير الحقيقية مثل $\sqrt{2}$ و π الخ لاتخزن بقيمتها الفعلية ولكن بتقريب حقيقى . فى واقع الأمر كثير من الارقام الحقيقية لها عدد محدد من الأرقام المعنوية فى الصورة الحقيقية إلا أنها لاتنتهى عند التعبير عنها فى الصورة الثنائية ، ومن ثم لاتخزن بقيمها الحقيقية . وتبعاً لذلك تظهر مشاكل جديدة فى رياضيات الحاسب . سنناقش بعضاً منها هنا (أنظر أيضاً المسألة ٢ – ١٧ و ٢ – ١٨) .

(أ) خطأ التحويل

تخزن معظم الحاسبات الأرقام فى شكلها الثنائى . وبذلك ، فالعدد 1/2 الذى يمثل فى النظام العشرى بـ 0.5 يمثل فى مثل هذه الحاسبات بـ 0.1 ولكن ، العدد الحقيقى 1/10 ليس له تمثيل ثنائى محدود .

1/10 = 0.0001100110011 . . . .

معتمداً على طول الكلمة (خلية الذاكرة) يمكن أن يكون خطأ التحويل بسيطا لايذكر . ولكن الخطأ سيتوالد كلما تضاعفت عدد العمليات الحسابية .

(ب) توالد خطأ التقريب

على سبيل المثال ، دعنا نفرض أن حاسبنا آلة عشرية تحتفظ بثلاثة خانات معنوية فقط . وبذلك يخزن الرقم 1/3 فى الحاسب بالصورة 0.333 .

اعتبر جمع 1/3 على نفسها 10 مرات . فنحصل على :

```
   .333
 + .333
   .666
 + .333
   .999
 + .333
  1.332
```

عند هذه النقطة فى الجمع ، يحتفظ الحاسب بـ 1.33 فقط حيث أننا افترضنا آلة ذات ثلاث خانات . وإذا أكملنا فنحصل على :

```
  1.33
 + .333
  1.66
 + .333
  1.99
 + .333
  2.32
 + .333
  2.65
 + .333
  2.98
 + .333
  3.31
```

حصلنا على 3.31 بدلا من أن نحصل على القيمة النظرية 3.33 . لقد توالد الخطأ بسرعة إلى حد ما. فى الحقيقة ، إذا جمعت 333. على نفسها 300 مرة فى الآلة الخاصة بنا سنحصل على 90.9 فقط بدلا من 99.9 المطلوبة . وعلاوة على ذلك ، إذا أضفت 333. إلى 100 فسوف يكون الناتج 100 مرة ثانية ، ومهما كانت عدد المرات التى تضيف فيها 333. إلى 100 ، فسوف تكون النتيجة دائماً 100 وبذلك فإضافة 333. إلى نفسها بصورة متتالية سيصل حتما إلى 100 فقط .

ومن الواضح ، أنه يمكن عمل شىء بخصوص هذه الحالة بالذات ، ولكن ستتولد الأخطاء إن لم نكن حريصين بخصوصها .

(حـ) عدم توافق (تزامل) الحسابات

أدرس تعبيرات الفورتران المكافئة للتعبيرات الرياضية .

$$(A + B) + C \quad \text{و} \quad A + (B + C)$$

نفرض أن دقة الحاسب ثمانى خانات معنوية ونفرض أن :

$$A = 0.30000000 \qquad B = 87654321. \qquad C = -87654322$$

وحيث أن ، الحاسب يحتفظ بثمان خانات فقط ،

$$A + B = 87654321.$$

وحيث أن A لا تؤثر فى نتيجة A + B . لذلك :

$$(A + B) + C = 87654321. - 87654322. = -1.0000000$$

ومن ناحية أخرى $B + C = -1.0000000$ ومن ثم :

$$A + (B + C) = 0.30000000 - 1.0000000 = -0.70000000$$

هناك ٣٠٪ فرق فى النتيجة ، وهذه قد تؤدى إلى فارق أكبر إذا استخدمت هذه القيم فى حسابات أخرى ، بمعنى أن خطأ التقريب يمكن أن يتزايد .

مسائل محلولة

الثـــوابت :

٢ – ١ أكتب الآتى كثوابت صحيحة بالفورتران .

(أ) 34,275	(د) 4.8×10^4	(ز) −7.3
(ب) +275	(هـ) −1,278000	(ح) 12
(جـ) −23.0	(و) 2.137×10^2	(ط) −138

(أ) 34275	(د) 4800	(ز) غير ممكنة – لأنه عدد غير صحيح
(ب) +275	(هـ) −1278000	(ح) 12
(جـ) −23	(و) غير ممكنة – لأنه عدد غير صحيح	(ط) −138

٢ – ٢ لماذا تعتبر الثوابت الصحيحة الآتية غير مقبولة فى الفورتران ؟

(أ) 2,371 (ب) −47.0 (جـ) 28E3 (د) 1234500000

(أ) تحتوى على فصلة (,)

(ب) تحتوى على علامة عشرية (.) .

(جـ) العدد الصحيح لايمكن كتابته فى الشكل الأسى .

(د) فى معظم الحاسبات ، لايمكن أن يتكون العدد الصحيح من أكثر من تسعة حروف .

٢ – ٣ أكتب الآتى كثوابت فورتران حقيقية فى الشكل المشروح سابقا وفى الشكل الأسى القياسى ،

(أ) 123	(د) 5.63×10^8	(ز) 0.123
(ب) 12.3	(هـ) −1,234,000	(ح) −0.356
(جـ) −3,400	(و) 0.00047	(ط) 3×10^{13}

(أ) 123. or .123E3
(ب) 12.3 or .123E2
(حـ) −3400. or −.34E4
(د) 5.63E−8 or .563E−7
(هـ) −1234000. or −.1234E7
(و) .000347 or .347E−3
(ز) 0.123 or 0.123E0
(ح) −.356 or −.356E0
(ط) 3.E13 or .3E14

أننا نؤكد أن الثابت (الحقيقى) يمكن أن يكتب بطرق مختلفة كشكل أسى ولكن بطريقة واحدة فقط فى الشكل الأسى القياسى . أى القوة 10 مضروبة فى أى رقم يتراوح ما بين 0.1 ، 1.0 أو بين 0.1 — ، 1.0 — .

٢ — ٤ لماذا تعتبر الثوابت الحقيقية الآتية غير مقبولة فى الفورتران ؟

(أ) 3,248.6
(ب) −13
(حـ) 1.6E128
(د) E+3
(هـ) 4E−2
(و) 4.3E3.7

(أ) يحتوى على فصلة (،)
(ب) ينقصه العلامة العشرية (.) .
(حـ) لايمكن أن يكون الأس أكثر من خانتين .
(د) لايمكن أن يظهر الأس بمفرده (يمكن أن تكتب 3 + 0.0E) .
(هـ) يجب أن تحتوى 4 على علامة عشرية .
(و) معظم الحاسبات تسمح فقط بالأس الصحيح .

المتغيرات

٢ — ٥ أذكر أياً من المتغيرات الصحيحة الآتية غير مقبولة فى الفورتران ، ولماذا ؟

(أ) NEXT
(ب) FIRST
(حـ) ALPHA
(د) L124
(هـ) J + 329
(و) MACBETH
(ز) 2I45
(ح) N(3)M

(أ) مقبولة .
(ب) غير مقبولة — الحرف الأول ليس I أو J أو K أو L أو M أو N .
(حـ) غير مقبولة — الحرف الأول ليس I أو J أو K أو L أو M أو N .
(د) مقبولة .
(هـ) غير مقبولة — غير مسموح بعلامة (+) .
(و) غير مقبولة — أكثر من ستة حروف .
(ز) غير مقبولة — يجب أن يكون أول حرف أبجدى .
(ح) غير مقبولة — يجب أن تكون الحروف أبجدية أو رقمية .

٢ — ٦ أذكر أياً من الثوابت الحقيقية الآتية غير مقبولة ، ولماذا ؟

(أ) ANSWER
(ب) LAMBDA
(حـ) ROOT2
(د) MAX
(هـ) ALTITUDE
(و) 4XYZ
(ز) STOP
(ح) X-RAY

(أ) مقبولة .

(ب) غير مقبولة – الحرف الأول L محجوز للمتغيرات الصحيحة .

(ﺣ) مقبولة .

(د) غير مقبولة – الحرف الأول M محجوز للمتغيرات الصحيحة .

(ﻫ) غير مقبولة – يجب أن يكون أول حرف أبجدياً .

(ز) غير مقبولة – لأنها كلمة دالة من كلمات الفورتران .

(ﺣ) غير مقبولة – الشرطة (–) ليست حرفاً أو رقماً .

التعبيرات الحسابية والعمليات :

٢ – ٧ بفرض أن التعبيرات ذات النمط المختلط غير مقبولة لدى الحاسب ، حدد أياً من التعبيرات الآتية مقبولة كتعبيرات فورتران وأوجد قيمة كل منها .

(أ) 2.4 + 3.85	(ﺣ) 3*4.5	(ﻫ) 3* + 5	(ز) 0 + 4.6
(ب) 12/5	(د) 3.**2	(و) 4.3 – 7.93	(ح) –7/3

(أ) 6.25

(ب) 2 (تم بتر الكسر فى القسمة الصحيحة) .

(ﺣ) غير مقبول – لأن حدود التعبير من نوعين مختلفين .

(د) 9 (يمكن أن يكون الأس عدد صحيحاً بالرغم من أن الأساس حقيقى)

(ﻫ) غير مقبولة – لايمكن أن تظهر العلامتين * و + متتاليتين .

(و) 3.63 –

(ز) غير مقبولة – لأن حدود التعبير من نوعين مختلفين .

(ح) 2 – (تم بتر الكسر فى القسمة الصحيحة) .

٢ – ٨ نفرض أن حسابات النمط المختلط مقبولة ، أوجد قيمة كل من التعبيرات التالية :

(أ) 3 + 4.8*2	(ﺣ) 2.*6/5	(ﻫ) 7/(4*2.)	(ز) 2*(3**2)
(ب) 5.2 + 12/8	(د) 2*6/5	(و) 7/(4*2)	(ح) 2*3**2

(أ) للضرب أسبقية على الجمع ، وبذلك :

4.8*2 تعطى 9.6 و 3 + 9.6 تعطى 12.6

(ب) القسمة الصحيحة 12/8 تعطى 1 و 1 + 5.2 تعطى 6.2 .

(ﺣ) 2.*6 تعطى الرقم الحقيقى .12 و 5/.12 تعطى الرقم الحقيقى 2.4 .

(د) 2*6 تعطى العدد الصحيح 12 والقسمة الصحيحة 12/5 تعطى 2 وتكون هى النتيجة

(ﻫ) 4*2. تعطى الرقم الحقيقى .8 و 7/8. تعطى 0.875

(و) 2*4 تعطى الرقم الصحيح 8 ، والقسمة الصحيحة 7/8 تعطى 0 وتكون هى النتيجة

(ز) 18

(ح) الرفع إلى الأس له أسبقية على الضرب ، ومن ثم تحصل على 18 كما فى (ز)

٢ – ٩ أكتب تعبير فورتران مناظر لكل من التعبيرات الرياضية التالية :

(أ) $(x + y)(u + v)$ (د) $a + \frac{b}{c^2}$ (و) $\frac{x^4}{4!}$

(ب) $3(x + y)$

(ﺣ) $3xy^2 - 2x^2y$ (ﻫ) $\frac{a + b}{c \cdot d}$

(أ) (X + Y)*(U + V) (ﺣ) 3.0*X*Y**2 – 2.0*X**2*Y (ﻫ) (A + B)/(C*D)

(ب) 3.0*(X + Y) (د) A + B/C**2 (و) X**4/(4.*3.*2.*1.)

٢ – ١٠ أوجد تعبيراً مكافئاً لكل من تعبيرات الفورتران التالية ، أى التعبير الذى لايغير ترتيب العمليات الحسابية ، وذلك بحذف الأقواس الزائدة .

(أ) (A*B)*(C + D) (ﺣ) (A*B*C)/((X*Y)**2)

(ب) (A*(B**2))/(C*D) (د) A*(B*C)

(أ) A*B*(C + D)

(ب) A*B**2/(C*D)

(ﺣ) A*B*C/(X*Y)**2

(د) A*(B*C) لايمكن حذف الأقواس (رغم أن A*B*C ، تحسب قيمتها بالصورة (A*B)*C، تكافىء رياضياً A*(B*C) ، إلا أن ترتيب العمليات مختلف وبذلك قد يؤدى خطأ التقريب إلى نتائج مختلفة . أنظر قسم ٢ – ١٠) .

٢ – ١١ أكتب تعبير فورتران مناظراً لكل من التعبيرات الرياضية التالية وذلك باستخدام الدوال الرياضية فى جدول ٢ – ٢ (صفحة ٣٥)

(أ) $\sqrt{a^2 + b^2}$ (ﺣ) $\frac{1}{|a \cdot b|} + c$

(ب) $e^{x+y} - \sin(x + ny)$ (د) $\cos(\log_{10}(a + 3b))$

(أ) SQRT(A**2 + B**2)

(ب) EXP(X + Y) – SIN(X + N*Y)

إذا كان الحاسب لا يقبل تعبيرات ذات نمط مختلط

EXP(X + Y) – SIN(X + FLOAT(N)*Y)

(ﺣ) 1./ABS(A*B) + C

(د) COS(ALOG10(A + 3.0*B))

٢ – ١٢ فيما يلى تعبيرات رياضية تناظرها تعبيرات فورتران غير صحيحة . أكتب تعبيرات الفورتران الصحيحة :

(أ) AB/CDE ، $\frac{a \cdot b}{c \cdot d \cdot e}$ (جـ) A/(B + C)**2 ، $\left(\frac{a}{b+c}\right)^2$

(ب) (X/Y)**N + 1 ، $\left(\frac{x}{y}\right)^{n+1}$ (د) SQRT(A**2/(B + C) ، $\sqrt{\frac{a^2}{b+c}}$

(أ) A*B/(C*D*E) (ب) (X/Y)**(N + 1) (جـ) (A/(B + C))**2 (د) SQRT(A**2/(B + C))

الجمل الحسابية

٢ – ١٣ إدرس الجمل الآتية :

(أ) A = B + C (جـ) A = A + A (هـ) AB = CD

(ب) B + C = A (د) A + B = C + D (و) A = ABS(A)

أى من الجمل السابقة مقبول كجمل فورتران ؟ وماهى الشروط الضمنية التى افترضناها لتلك الجمل المقبولة ؟

تذكر أن المتغير ، أى اسم مكان التخزين فقط ، هو الذى يمكن أن يظهر على يسار علامة التساوى .

(أ) مقبولة .

(ب) غير مقبولة حيث B + C ليست متغيراً .

(جـ) مقبولة .

(د) غير مقبولة حيث A + B ليست متغيراً .

(هـ) مقبولة .

(و) مقبولة .

نحن نفترض ضمنياً أن كل المتغيرات على الجانب الأيمن من علامة التساوى قد تم تعريفها مسبقاً ، أى أن الأرقام موجودة فعلا فى أماكن التخزين التى تحمل هذه الأسماء .

٢ – ١٤ نفرض أن A و B و J و K تحتوى على القيم الآتية : A = 2.7 و B = 3.5 و J = 3 و K = −2 . أوجد قيم X و L بعد كل زوج من الجمل التالية :

(أ) X = A + J*K**2 + B
L = A + J*K**2 + B

(جـ) X = K/3*A/2
L = K/3*A/2

(ب) X = 5*J/4*K
L = 5*J/4.0*K

(د) X = ABS(A − J*B)/5
L = ABS(A − J*B)/5

(أ) عند إيجاد قيمة التعبير يقوم الحاسب بتنفيذ رفع الأس أولا ، ثم بعد ذلك الضرب وأخيراً الجمع .

$$A + J*K**2 + B = 2.7 + 3(-2)^2 + 3.5 = 2.7 + 12 + 3.5 = 18.2$$

يخزن الحاسب 18.2 فى X حيث أن X متغير حقيقى ، ولكن حيث أن L متغير صحيح ، فيخزن الحاسب 18 فى L (أى ، الجزء الصحيح فقط من 18.2) أى أن X = 18.2 و L = 18

(ب) ينفذ الحاسب الضرب والقسمة من اليسار إلى اليمين :

5*J/4*K = 15/4*K = 3*K = −6
5*J/4.0*K = 15/4.0*K = 3.75*K = −7.5

وبسبب القسمة الصحيحة ، فإن 15/4 تعطى 3 ولكن 15/4.0 تعطى 3.75 وحيث أن X حقيقية ، فتتحول 6 — إلى رقم حقيق 6.0 — ثم تخزن بعد ذلك فى X وحيث أن L متغير صحيح ، فقد تم بتر 7.5 — وأصبحت 7 — ثم بعد ذلك تم تخزينها فى L . وبذلك أصبحت X = — 6.0 و L = — 7

(ح) بسبب القسمة الصحيحة ، فإن K/3 تعطى 0 ومن ثم X = 0.0 و L = 0

(د) ABS(A − J*B)/5 = |2.7 − 3(3.5)|/5 = |−7.8|/5 = 1.56. So X = 1.56 and L = 1

٢ — ١٥ افرض أن A و B لها القيم التالية A = 2.5 و B = 3.5 أوجد قيم A و B بعد تنفيذ كل من مجموعات الجمل التالية :

(أ) A = B
B = A

(ب) T = A
A = B
B = T

(أ) توجه الجملة A = B الحاسب بأن يمسح القيمة الحالية الموجودة فى A ويعوض عنها بالقيمة الحالية لـ B ومن ثم .

A ← ~~2.5~~ 3.5

وتوجه الجملة التالية B = A الحاسب بأن يمسح القيمة الحالية لـ B (ويعوض) عنها بالقيمة الحالية لـ A ومن ثم .

B ← ~~3.5~~ 3.5

لاحظ أن قيمة B لم تتغير .

(ب) تعطى الجمل الثلاث .

T ← 2.5, A ← ~~2.5~~ 3.5, B ← ~~3.5~~ 2.5

لاحظ أن قيم A و B قد تم تبديلها عن طريق تقديم مكان تخزين مساعد ليحتفظ بقيمة أحد المتغيرات خلال عملية التبديل .

٢ — ١٦ افرض أن المتغيرات A و B و C قد تم تعريفها مسبقاً . اكتب جملة الفورتران التى (أ) تضاعف قيمة A (ب) تزيد قيمة B بأربعة (ج) تقلل قيمة C بالقيمة الموجودة فى A (د) تخزن القيمة المتوسطة لـ A و B و C فى AVE أى مجموع A و B و C مقسوماً على 3. (هـ) تخزن الجذر التربيعى لمجموع مربعات A و B و C فى DIST أى $\sqrt{A^2 + B^2 + C^2}$

(١) A = 2.0*A
(ب) B = B + 4.0
(ج) C = C − A
(د) AVE = (A + B + C)/3.0
(هـ) DIST = SQRT(A**2 + B**2 + C**2)

مسائل متنوعة :

٢ — ١٧ ناقش الطرق الثلاث الآتية لكتابة x^2 فى الفورتران :

(أ) X*X (ب) X**2 (ج) X**2.0

فى (ا) تكتب x^2 بدلالة الضرب وسوف يتم حساب قيمتها بهذه الطريقة فى (ب) ستحسب قيمة X**2 بالضرب كنتيجة لترجمة X**2 إلى X*X فى (ج) الأس حقيق وبذلك فإن (ج) ستحسب قيمتها EXP(2.0 | ALOG(X)) بشرط أن تكون X موجبة . (انظر قسم ٢ — ٨) ويتضح أن (ا) و (ب) أفضل من (ج) . فى الحقيقة يمكن أن تعطى (ج) نتيجة مختلفة نسبياً فى ضوء خطأ التقريب عند حساب قيمة EXP و ALOG وأكثر من ذلك ، فبالرغم أن (ا) و (ب) تتطلب نفس وقت التشغيل ، فإن (ا) لا تتطلب ترجمة إضافية .

٢ — ١٨ ناقش جزئى البرنامج التاليين :

(أ) XN = N
AVE = SUM/XN

(ب) AVE = SUM/FLOAT(N)

(تنبه أن كليهما مكتوب لتجنب قسمة النمط المختلط)

فى (ا) تحول الرقم الصحيح N أولا إلى رقم حقيقى ويخزن فى XN ثم نحسب المتوسط SUM/XN لاحظ أن ذلك يتطلب مكان تخزين إضافى . ومن وجهة أخرى ، فإن (ب) تتطلب جملة واحدة حيث أننا نستخدم الدالة FLOAT فى الحسابات . وهكذا ، فتبدو (ب) أكثر فائدة من (ا) . ولكن ، إذا كان البرنامج طويلا ، ونحتاج إلى تحويل N إلى رقم حقيقى عدة مرات ، لكان من الأحسن تحويل N مرة واحدة إلى رقم حقيقى بواسطة XN = N بدلا من استخدام FLOAT (N) طوال البرنامج .

مسائل تكميلية

الثوابت :

٢ - ١٩ اكتب ما يلى فى صورة ثوابت فورتران صحيحة .

(ا) 2,348	(د) 3.1214×10^3	(ز) 23.51
(ب) 5.31×10^3	(هـ) −37.0	(ح) 1,250,000
(جـ) −531	(و) $21,500 \times 10^{-3}$	(ط) $57,000.0 \times 10^{-2}$

٢ - ٢٠ لماذا تعتبر ثوابت الفورتران الصحيحة التالية غير مقبولة ؟

(ا) 2,578	(ب) −784.0	(جـ) 37810000000	(د) 38E+2

٢ - ٢١ اكتب القيم التالية فى شكل ثوابت فورتران حقيقية وأيضاً فى الشكل الأسى القياسى .

(ا) 2,345	(جـ) 2,348,500	(هـ) .00005829
(ب) −1.63	(د) -7.63×10^{-5}	(و) 21×10^{10}

٢ - ٢٢ لماذا تعتبر ما يلى ثوابت فورتران حقيقية غير مقبولة ؟

(ا) −3.6E134	(جـ) 52E−7	(هـ) E21
(ب) 2,356 4	(د) 256	(و) −1,378.0

٢ - ٢٣ حدد أى زوج من الثوابت يمثل نفس العدد .

(ا) 4.36E01	43.6	(د) +12345	12345
(ب) 5.43E+02	543	(هـ) 234.E4	0.234E6
(جـ) 4.0E − 5	0.00004	(و) 1.	1.00

المتغيرات :

٢ - ٢٤ ادرس قائمة الأسماء التالية . وحدد من بينها ما هو مقبول إما (١) كمتغيرات صحيحة أو (٢) كمتغيرات حقيقية . اذكر السبب فى أن بقية أسماء المتغيرات غير مقبولة ؟

(ا) MORE	(و) A567B	(ك) AMOUNT
(ب) LESS	(ز) RATE	(ل) A + 567
(جـ) NEITHER	(ح) X34.7	(م) A1234S
(د) AAAAAA	(ط) GAMMA	(ن) IN-OUT
(هـ) SPQR	(ى) KAPPA	(ص) OPERAND

العمليات والمتغيرات الرياضية

٢ – ٢٥ بفرض أن الحاسب لا يقبل حساب النمط المختلط . حدد مما يلى ما هو مقبول كتعبيرات فورتران وأوجد قيمته . عرف وحدد الأخطاء فى التعبيرات غير المقبولة .

(أ) 6.3 + 5.2E3	(جـ) 6.*4	(هـ) 0/3.4	(ز) 9/ – 4
(ب) - 18/7	(د) 6.**4	(و) 6+ – 7	(ح) –4/9

٢ – ٢٦ افرض أن الحاسب يقبل فعلا حساب النمط المختلط . أوجد قيمة كل تعبير مما يلى :

(أ) 3 – 5*2.5	(جـ) 19/(2*5)	(هـ) –3*2**3	(ز) (7/3)/(6/5)
(ب) 2.8 – 17/5	(د) 19/(2.*5)	(و) –3*(2**3)	(ح) 2**2*2/3

٢ – ٢٧ اكتب تعبير فورتران مناظراً لكل تعبير رياضى مما يلى :

(أ) $(2x + y)(3z - 4w)$

(ب) $4x^2y - 3xy + 7yz^3$

(جـ) $\left(\frac{a}{c} + \frac{b}{d}\right)^3$

(د) $\frac{x^3}{5!}$

(هـ) $\left(3 + \frac{a}{b}\right)^{m+1}$

(و) $\dfrac{\frac{a}{b} + 6}{x - \frac{y}{z}}$

٢ – ٢٨ أوجد التعبير المكافىء لكل تعبير من تعبيرات الفورتران الآتية . أى التعبير الذى لا يغير ترتيب العمليات الحسابية بحذف الأقواس غير اللازمة (الزائدة) .

(أ) (X + Y) + Z	(جـ) A + ((B*C)/D)	(هـ) (X*(Y – Z))*(A**2)
(ب) X + (Y + Z)	(د) (A + (B**3))/(X*Y)	(و) ((A**2) + (B**2)) – (D*(E/F))

٢ – ٢٩ اكتب تعبيرات الفورتران المناظرة لكل من التعبيرات الرياضية التالية باستخدام الدوال الرياضية الموضحة فى جدول ٢ – ٢ (صفحة ٣٥) .

(أ) $\log_e(x + y)^2$

(ب) $\log_{10}(a - b)^2$

(جـ) $\left|\sqrt{x^2 + y^2} - \frac{z^2}{\cos(a + b)}\right|$

(د) $\sqrt{|\sin(a - |b|)|}$

(هـ) $\sqrt{5x^2 + 8y^2}$

(و) $\sin(x - 2y) + e^{xy} - |x^2 - y^2|$

(د) $e^{|a|} - \frac{b^2}{|c|}$

(ح) $\sqrt{|\cos(a - nb)|}$

٢ – ٣٠ فيما يلى تعبيرات رياضية وتعبيرات فورتران غير صحيحة . اكتب تعبيرات الفورتران الصحيحة .

(أ)	$a + \frac{b}{c \cdot d}$	A + B/CD
(ب)	$\frac{x^{n+1}}{y^{n-1}}$	(X**N + 1)/(Y**N – 1)
(جـ)	$\sin(x + n)$	SIN(X + FLOAT(N)
(د)	$\log_{10}\lvert a \cdot b\rvert$	LOG(ABS(AB)

الجمل الحسابية

٢ – ٢١ أى من جمل الفورتران الآتية مقبولة ؟

(ا) X = Y + Z
(ب) X + Y = Z
(جـ) Y + Z = X
(د) XY = ZW
(هـ) XY = Z*W
(و) X*Y = Z*W
(ز) ABC = DEF
(ح) ABC = 2DEF
(ط) A*B*C = D*E*F

٢ – ٢٢ اعتبر أن X و Y و L و M تحتوى على القيم التالية X = 3.1 و Y = 4.6 و L = 2 و M = 3 أوجد القيمه النهائية لكل من A و J بعد تنفيذ مجموعات الجمل التالية :

(ا)
A = X – 2*Y + X/L**2
J = X – 2*Y + (X/L)**2

(ب)
A = 8/M + 3*L
J = 8.0/M – Y

(جـ)
A = L/M*X/Y
J = L/(M + X)*Y

(د)
A = X + 2*Y
A = 2*A + 4
J = A

(هـ)
J = L + 3*M
J = J**2 + Y
A = X + J

(و)
J = SQRT(X*Y)
A = J*Y
J = A*M
A = A + ABS(FLOAT(J))

٢ – ٢٣ اعتبر أن X و Y و Z قد تم تعريفها مسبقاً . اكتب الجملة الحسابية التى تنجز الآتى : (أ) تزيد قيمة X بـ 3.2 (ب) تجعل قيمة Y ثلاثة أضعاف ، (جـ) تربيع قيمة Z (د) تخزن حاصل ضرب القيم X و Y و Z فى PRDT (هـ) تخزن طول الوتر لمثلث قائم الزاوية فى HYP وأضلاعه لها الأطوال X و Y (و) تخزن القيمة المتوسطة لكل من X و Y و Z فى AVE .

٢ – ٢٤ افرض أن A و B و C قد تم تعريفها مسبقاً . اكتب جزء البرنامج الذى يبدل قيم A و B و C بحيث تأخذ A قيمة B وتأخذ B قيمة C وتأخذ C قيمة A (تلميح : انظر مسألة ٢ – ١٥ (ب)) .

اجابات للمسائل التكميلية المختارة

٢ - ١٩ (أ) 234 (ب) 5310 (جـ) 531 — (د) مستحيل (هـ) 37 — (و) مستحيل
(ز) مستحيل (ح) 1250000 (ط) 570

٢ - ٢٠ (أ) تحتوى على فصلة (،)
(ب) تحتوى على علامة عشرية (.)
(جـ) لا يمكن أن يكون العدد الصحيح فى معظم الحاسبات أكثر من تسعة حروف .
(د) لا يمكن أن يكتب العدد الصحيح فى الشكل الأسى .

٢ - ٢١ (أ) 0.2345E4 أو .2345
(ب) 0.163E1– أو 1.63–
(جـ) 0.23485E7 أو .2348500
(د) 0.763E-4– أو 7.63E-5–
(هـ) 0.5829E-4 أو 0.00005829
(و) 0.21E20 أو 21.E18

٢ - ٢٢ (أ) لا يمكن أن يكون الأس أكثر من رقمين فى معظم الحاسبات .
(ب) يحتوى على فصلة (،) .
(جـ) تنقصه العلامة العشرية (.) .
(د) تنقصه العلامة العشرية (.) .
(هـ) لا يمكن أن يظهر الأس بمفرده . (يمكن أن يكتب 0.0E21)
(و) تحتوى على فصلة (،) .

٢ - ٢٣ (أ) نعم . (ب) ليس أى منهما عدداً صحيحاً . (جـ) نعم . (د) . نعم . (هـ) لا . (و) نعم .

٢ - ٢٤ (أ) عدد صحيح (ب) عدد صحيح . (جـ) أكثر من ستة حروف (د) حقيق
(هـ) يبدأ برقم (و) حقيق (ز) حقيق (ح) غير مسموح بالعلامة العشرية
(ط) حقيق (ى) عدد صحيح (ك) حقيقية (ل) غير مسموح بعلامة +
(م) حقيق (ن) غير مسموح بالشرطة (–) (ص) أكثر من ستة حروف .

٢ - ٢٥ (أ) 5206.3 (ب) 2 –– (جـ) غير مقبول (د) .1296 (هـ) غير مقبول . (و) غير مقبول .
حيث لا يمكن أن تظهر * و – متتاليين (ز) 2 — (ح) 0

٢ - ٢٦ (أ) 9.5–· (ب) 0.2 — (جـ) 1 (د) 1.9 (هـ) 24 — (و) 24–– (ز) 2 (ح) 2

٢ - ٢٧ (أ) (2.*X + Y)*(3.*Z – 4.*W)
(ب) 4.*X**2*Y – 3.*X*Y + 7.*Y*Z**3
(جـ) ((A + B)/(C + D))**3
(د) X**5/(5.*4.*3.*2.*1.)
(هـ) (3. + A/B)**(M – 1)
(و) (A/B + 6.)/(X – Y/Z)

٢ - ٢٨ (أ) X + Y + Z
(ب) X + (Y + Z)
(جـ) A + B*C/D
(د) (A + B**3)/(X*Y)
(هـ) X*(Y – Z)*A**2
(و) A**2 + B**2 – D*(E/F)

٢ - ٢٩ (أ) SQRT(5.*X**2 + 8.*Y**2)
(ب) SIN(X – 2.*Y) + EXP(X*Y) – ABS(X**2 – Y**2)
(جـ) EXP(ABS(A)) – B**2/ABS(C)
(د) SQRT(ABS(COS(A – FLOAT(N)*B)))
(هـ) ALOG((X + Y)**2)
(و) ALOG10((A – B)**2)
(ز) ABS(SQRT(X – Y**3) – Z**3/COS(A + B))
(ح) SQRT(ABS(SIN(A – ABS(B))))

٢ - ٣٠ (أ) A + B/(C*D)
(ب) (X**(N + 1))/(Y**(N – 1)) or X**(N + 1)/Y**(N – 1)
(جـ) SIN(X + FLOAT(N))
(د) ALOG10(ABS(A*B))

٢ - ٣١ (أ) نعم . (ب) لا . (جـ) لا . (د) نعم . (هـ) نعم . (و) لا . (ز) نعم . (ح) لا . (ط) لا .

٢ - ٣٢ (أ) A = –5.325, J = –3
(ب) A = 4.0, J = –7
(جـ) A = 0.0, J = 92
(د) A = 28.6, J = 28
(هـ) A = 56.1, J = 53
(و) A = 54.8, J = –41

٢ - ٣٣ (أ) X = X + 3.2
(ب) Y = 3.*Y
(جـ) Z = Z**2
(د) PRDT = X*Y*Z
(هـ) HYP = SQRT(X**2 + Y**2)
(و) AVE = (X + Y + Z)/3.0

٢ - ٣٤

```
T = A
A = B
B = C
C = T
```

الفصل الثالث

الادخال / الاخــراج العــددى

٣ ـــ ١ مقــــدمة

سنرى فى هذا الفصل ، كيف يقرأ الحاسب البيانات الخارجية (المدخلات) ويطبع البيانات (المخرجات) يتم هذا عن طريق جملتى READ و WRITE على الترتيب . وتسمى هذه عمليات الإدخال/ الإخراج أو ببساطة عمليات (I/O) . وكما ذكرنا مسبقاً فسنعتبر أن وحدة الإدخال هى قارىء البطاقات ، ووحدة الإخراج هى آلة الطباعة (إلا إذا نص أو تضمن على غير ذلك) .

تأمر جملة READ الحاسب بأن يقرأ معلومات عددية أو رمزية من مجموعة بطاقات (مجموعة بطاقات البيانات) . وتأمر جملة WRITE الحاسب بأن يطبع معلومات على آلة الطباعة . عادة ما تصاحب جملة READ أو WRITE جملة FORMAT. فى حالة الإدخال فان جملة FORMAT تمد الحاسب بنوع المعلومات والأماكن الخاصة بها فى سجل إدخال . فى حالة الإخراج ، نجد أن جملة FORMAT تمد الحاسب أيضاً بمكان طباعة المعلومات . ومن الواضح أن جملة FORMAT يمكن أن تكون مفصلة جداً . ولكن كثيراً من الحاسبات تسمح بما يسمى ملامح الإدخال ـ الإخراج غير المصاغ . سنناقش أولا مثل هذا (I/O) غير المصاغ . (تستخدم بعض الكتب المصطلح « صياغة غير مقيدة » فضلا عن المصطلح « غير مصاغ ») .

٣ ـــ ٢ ادخال / اخراج غير مصاغ

بعد قراءة هذا القسم ، يمكن القارىء أن يمضى إلى قسم ٣ — ١٠ ، والفصول الثلاثة التالية قبل العودة إلى بقية هذا الفصل (إلا إذا كانت إمكانيات الحاسب المتاح للقارىء لا تسمح بادخال ـ إخراج غير مصاغ) .

ويمكن أن يختلف (I/O) غير المصاغ ، والذى يتم مناقشته فى هذا الفصل ، من جهاز إلى آخر باختلافات طفيفة . ولذا ، ننصح القارىء بالرجوع إلى التفصيلات الخاصة بالحاسب المتاح له .

فيما يلى جملة PRINT غير المصاغة النموذجية :

```
PRINT, A, B, C, AREA, M, N
```

لاحظ أن كلمة PRINT هى وكل متغير فيما عدا المتغير الأخير متبوعة بفصلة . تأمر هذه الجملة الحاسب بأن يطبع قيم المتغيرات A ، B ، C ، AREA ، M ، N بهذا الترتيب المعطى . ستطبع القيم بشكل مخصص (معروف) مسبقاً ، وسوف تطبع الأرقام الحقيقية عادة فى الشكل الأسى. يجب تحديد عدد المتغيرات التى يمكن أن تطبع فى السطر الواحد أيضاً مسبقاً . بينما سيبدأ الحاسب الطباعة على سطر جديد كلما قابل جملة PRINT .

فيما يلى جملة READ غير المصاغة النموذجية :

```
READ, A, B, C, LOT, AREA
```

لاحظ أن كلمة READ وكل متغير فيما عدا الأخير متبوعة بفصلة . تأمر هذه الجملة الحاسب بأن « يقرأ » خمسة أرقام . رقم لكل متغير (مبتدأ) من أول بطاقة فى مجموعة البيانات . إذا كان هناك خمسة أرقام أو أكثر على هذه البطاقة ، فسوف تخصص أول خمس قيم إلى A و B و C و LOT و AREA على الترتيب ، وتهمل البطاقة بعد ذلك . ولكن إذا كان هناك أقل من خمسة أرقام على البطاقة فسيواصل الحاسب القراءة من مجموعة البيانات إلى أن يجد الأرقام المطلوبة جميعها . أى أن ، تنفيذ جملة READ لا يكتمل إلا إذا خصصت قيم لكل المتغيرات المذكورة فى القائمة .

لاحظ أن بطاقة البيانات تهمل بعد أن تستخدم فى القراءة . ومن ثم تكون دائماً البطاقة التى لم تقرأ بعد على قمة مجموعة البيانات .

يجب أن تفصل الأرقام التى تقرأ من مجموعة بطاقات البيانات بجملة إدخال غير مصاغة بفصلات (تسمح بعض مراكز الحاسبات بأن تكون الأرقام مفصولة بواسطة مسافة فارغة (خالية) بدلا من الفصلة) . تثقب الأرقام الحقيقية أما فى الشكل العشرى أو فى الشكل الأسى ومن الواضح انه يجب أن تكون الأرقام الموجودة على البطاقة (بطاقات) هى والمتغيرات المناظرة لها من نفس النوع .

مثال ٣ — ١

افترض أن حاسباً ينفذ الجملتين التاليتين :

```
READ, I, J, K
READ, L, M
```

واعتبر أن بطاقات البيانات قد تم تثقيبها كالآتى :

أول بطاقة بيانات :	222,76
ثانى بطاقة بيانات :	38,175,55
ثالث بطاقة بيانات :	194,58,567

حين ينفذ الحاسب جملة READ الأولى يخصص 222 و 76 إلى I و J على الترتيب وبعد ذلك يذهب إلى بطاقة البيانات التالية ويخصص 38 إلى K ويكمل هذا تنفيذ جملة READ الأولى وتهمل بطاقة البيانات الثانية ، رغم أن الأرقام 175 و 55 لم تقرأ . حين ينفذ الحاسب جملة READ الثانية ، سيقرأ البطاقة التى على قمة مجموعة بطاقات البيانات . حينئذ يستخدم الحاسب بطاقة البيانات الثالثة لتخصيص 194 إلى L و 58 إلى M وعلى ذلك يكون لدينا

I = 222, J = 76, K = 38, L = 194, M = 58

تتطلب بعض التسهيلات الحسابية أن تكون الأرقام على بطاقات البيانات مفصولة بواسطة مسافات خالية (يرمز إليها بـ b كدليل) بدلا من الفصلات (,) . فى هذه الحالات يجب أن تثقب البيانات كالتالى :

أول بطاقة بيانات :	222_b76
ثانى بطاقة بيانات :	38_b175_b55
ثالث بطاقة بيانات :	$194_b\ 58_b567_b$

٣ — ٣ مقدمة للادخال / الاخراج المصاغ

تم تقديم جمل الإدخال - الإخراج غير المصاغ (الصياغة غير المقيدة) أولا ، حتى يتمكن القراء من البدء فى كتابة البرامج والتعامل مع الحاسب فى أقرب وقت ممكن . وبهذا لتجنب التأخير حتى نلم بتفاصيل كتابة جمل FORMAT . ومع ذلك ، فلأن I/O غير

المصاغ تنفذ فعلا بصيغ سبق تحديدها ، فلا نملك تحكما كبيراً فى كيفية قراءة وطباعة المعلومات . نناقش هنا تفاصيل كتابة جمل I/O المصاغة وسنقصر مناقشتنا على I/O الخاص بالقيم الرقمية . وسنعامل الأنواع الأخرى فى فصل لاحق .

فى جهاز حاسب كبير يمكن استخدام وحدات كبيرة للادخال والإخراج . وللتفرقة بين هذه الوحدات ، يخصص لهم أرقام – يوضح جدول ٣ – ١ أرقام وحدات نمطية .

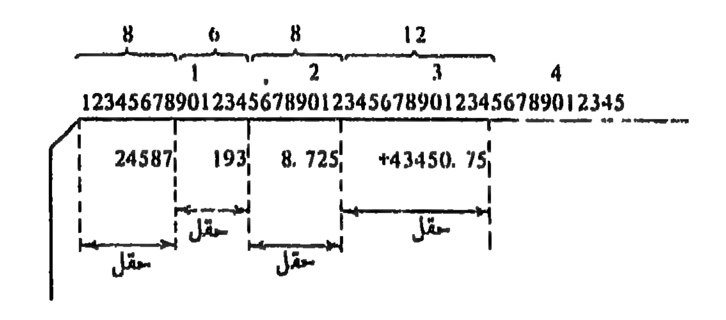
جدول ٣ – ١

الوحدة	الرقم
وحدة الشرائط	١
قارئ البطاقات	٥
طابعة سطرية	٦
مثقب البطاقات	٧

(تم استخدام هذه الأرقام أولا فى كثير من حاسبات IBM وما تزال منتشرة الاستخدام) فى هذا الكتاب ، نفرض أن وحدة الإدخال الخاصة بنا وهى قارئ البطاقات ، وتحمل رقم 5 ووحدة الإخراج الخاصة بنا وهى آلة الطباعة وتحمل رقم 6 .

حيث أن ، وحدة الإدخال هى قارئ البطاقات ، فستعطى بيانات الإدخال إلى الحاسب عن طريق مجموعة بطاقات . وهى تشبه البطاقات التى تثقب عليها جمل الفورتران تماماً . إلا أن قوانين تثقيب جمل الفورتران لا تنطبق هنا ، أى ، ممكن أن نستخدم الثمانين عموداً بالكامل من بطاقة البيانات ، وليس لأى عمود غرض معين كما فى عمود 6 مثلا .

تثقب المدخلات فى مجموعة من الأعمدة المتلاصقة على بطاقة البيانات . يوضح شكل ٣ – ١ بطاقة بيانات مثقب عليها أرقام ، الرقم الأول 24587 مثقب فى الأعمدة من 1 إلى 8 الرقم الثانى 193 مثقب فى الأعمدة من 9 إلى 14 الرقم الثالث 8.725 مثقب فى الأعمدة من 15 إلى 22 والرقم الرابع 43450.75 + مثقب فى الأعمدة من 23 إلى 34 وتسمى مجموعة الأعمدة المتلاصقة هذه حقول ، ويسمى عدد الأعمدة فى الحقول بعرض الحقل . لاحظ أن كل رقم مثقب فى حقله مضبط – جهة اليمين (*right-justified*) أى تظهر آخر خانة من الرقم فى العمود الأخير من حقله . (تستخدم بعض النصوص المصطلح منتظم – يمينى بدلا من المصطلح مضبط جهة اليمين).



شكل ٣ – ١ بطاقة بيانات توضح الحقول وعرض الحقول .

افترض أن الأرقام الأربعة السابقة ستخصص للمتغيرات ID و LOT و RATE و PRICE على الترتيب . يتم هذا فى الفورتران بواسطة جملة READ التى تصاحبها جملة FORMAT والتى قد تكتب كالتالى :

```
      READ(5, 100) ID, LOT, RATE, PRICE
100   FORMAT(I8, I6, F8.0, F12.0)
```

بدراسة الرقين بين القوسين اللذين يتبعان كلمة READ نجد أن الرقم الأول 5 هو رقم وحدة الإدخال يشير إلى قارئ البطاقات (كنا سنكتب 1 بدلا من 5 إذا أردنا استخدام وحدة الشرائط بدلا من قارئ البطاقات ، إذا كانت بيانات الإدخال مخزنة على شرائط ممغنطة مثلا) ونجد أن الرقم الثانى 100 يشير إلى رقم جملة FORMAT المصاحبة لجملة READ هذه (تمدنا جملة FORMAT بالمعلومات الضرورية عن نوع ومكان المتغيرات) . إن الرقم 100 هذا اختيارى تماماً. تأتى بعد ذلك أسماء المتغيرات ويتم ذكرها بالترتيب الذى تثقب به القيم على بطاقة البيانات .

ينطبق هذا الترتيب أيضاً على البنود الأربعة فى جملة FORMAT المصاحبة وهى I8 و I6 و F8.0 و F12.0 . تسمى هذه البنود مواصفات الحقل (أحياناً موصف الحقل) أو كود الصيغة . يحتوى كل مدخل على حرف يدل على نوع البيانات ، إذا كان صحيحاً أو حقيقياً مثلا ، ورقم يدل على عرض الحقل ، ومن ثم موقع الحقل لعنصر البيانات المناظرة . فى حالتنا هذه ، تدل مواصفات الحقل الأول I8 على أن الرقم الأول صحيح ويمكن أن يوجد فى أول 8 أعمدة من البطاقة . وتدل مواصفات الحقل الثانى I6 على أن الرقم الثانى أ سا صحيح ويمكن أن يوجد فى الأعمدة الستة التالية من البطاقة وتدل مواصفات الحقل الثالث F8.0 على أن الرقم الثالث حقيق ويمكن أن يوجد فى الأعمدة الثمانية التالية من البطاقة وتدل المواصفات الرابعة F12.0 على أن الرقم الرابع حقيق ويمكن أن يوجد فى الإثنى عشر عموداً التالية من البطاقة .

فى هذا المثال ، تأخذ مواصفات حقل الرقم الصحيح الصيغة *Iw* ومواصفات حقل الرقم الحقيقى الصيغة *Fw*.0 حيث تشير *w* إلى عرض الحقل . بفرض أن الأرقام الحقيقية تثقب بالصيغة العشرية .

٣ — ٤ الادخال المصاغ ومواصفات حقل الادخال

يقرأ الحاسب البيانات (المدخلات) بواسطة جملة READ تصاحبها جملة FORMAT كما تم توضيحه فى الجزء الأخير ويكون الشكل العام لمثل هاتين الجملتين كما يلى :

```
      READ(m, n) variable list
n     FORMAT(format list)
```

بالتحديد تبدأ جملة READ بكلمة READ يتبعها عدد صحيح *m* يحدد رقم وحدة الإدخال وعدد صحيح *n* يحدد جملة FORMAT . تفصل بين *m* و *n* فصلة ويحاط الرقين بقوسين . تحتوى قائمة المتغيرات التى تلى القوس أسماء أماكن الذاكرة التى تخزن فيها البيانات مفصولة عن بعضها بفصلات .

تبدأ جملة FORMAT المصاحبة والتى تحمل رقم الجملة *n* بكلمة FORMAT تتبعها قائمة بمواصفات الحقول (كود الصيغة) مفصولة عن بعضها بفصلات ، وتحاط القائمة بقوسين . يجب أن تتفق مواصفات الحقول مع قائمة المتغيرات ليس فقط فى الترتيب ولكن أيضاً فى النوع .

رغم أن جمل FORMAT تستخدم مع كل من جمل الإدخال والإخراج إلا أننا سنناقش الآن معنى مواصفات الحقول حينما تستخدم فى جمل الإدخال فقط . (وسوف نناقش معناها مع جمل الإخراج فى قسم ٣ — ٦) .

تحتوى مواصفات الحقول فى جملة FORMAT على عرض الحقول فى المدخلات بالترتيب وهى تحدد بطريقة منفردة مجموعة من الحقول المتتالية على بطاقة البيانات كما يلى :

يمثل أول حقل أول عدد w_1 من الأعمدة ، وثانى حقل عدد w_2 من الأعمدة التالية وهكذا . ولقد كانت عروض الحقول فى مثال القسم السابق 8 ، 6 ، 8 و 12 على الترتيب .

لاحظ أنه فى حالة القراءة بمواصفات حقول رقمية (التى سوف نناقشها فيما بعد) تطبق القاعدتان :

١ - تعتبر المسافات الخالية كأصفار .

٢ - تأخذ القيم التى لا تحمل إشارة على أنها موجبة .

إذا استخدمنا مواصفات حقول غير رقمية ، فيمكن أن نستثنى القاعدة الأولى .

حقل I :

مواصفات الحقل العامة لقراءة عدد صحيح هى

Iw

حيث يشير الحرف *I* إلى أن المتغير من النوع الصحيح ، ويشير *W* إلى رقم ثابت صحيح بدون إشارة يعطى عرض الحقل .

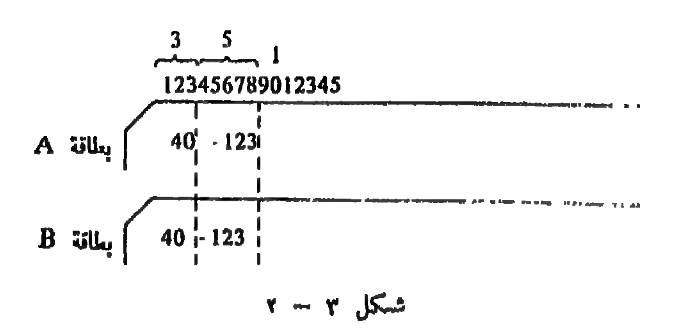
بما أن المسافات الخالية تعتبر أصفاراً فى الحقول الرقمية فمن الضرورى أن يثقب العدد الصحيح مضبطاً من الطرف الأيمن (right-justilied) فى حقله بحيث تظهر آخر خانة فى العدد الصحيح فى آخر عمود من الحقل المعطى .

إدرس الجمل التالية :

```
     READ(5, 20) M, N
20   FORMAT(I3, I5)
```

تأمر هذه الجمل الحاسب بأن يقرأ عددين صحيحين من بطاقة البيانات ويخزنها فى M و N على الترتيب . مواصفات M هى I3 ومواصفات N هى I5 . حيث أن عروض الحقول هى 3 و 5 على الترتيب يخصص إلى M العدد الصحيح الموجود فى الأعمدة 1 إلى 3 ويخصص إلى N العدد الصحيح الموجود فى الأعمدة 4 إلى 8 .

افرض أن العددين الصحيحين 40 و 123 مطلوب تخزينهما فى M و N على الترتيب حينئذ يجب أن تثقب بطاقة البيانات كما فى البطاقة (A) من الشكل ٣ - ٢ . ومن ناحية أخرى ، لو وجب تثقيب بطاقة البيانات كما فى البطاقة (B) من الشكل ٣ - ٢ لكانت جملة READ السابقة تخصص 400 إلى M و 1230 - إلى N حيث أن المسافات الخالية فى الأعمدة 3 و 8 تترجم كأصفار .



شكل ٣ - ٢

حقل F

تقرأ الثوابت الحقيقية فى الشكل العشرى باستخدام حقل *F* . الشكل العام هو

Fw.d

تشير *F* إلى أن البيانات من النوع الحقيقى وأن الرقم مكتوب فى الشكل العشرى ، حيث تشير *w* إلى عرض الحقل وتمثل بثابت صحيح

بدون إشارة ، وتمثل *d* عدد الخانات العشرية وهى عدد صحيح بدون إشارة أيضا . ومع ذلك ، يمكن أن تهمل *d* فى المدخلات إذا كان العدد المثقب فى الحقل به علامة عشرية .

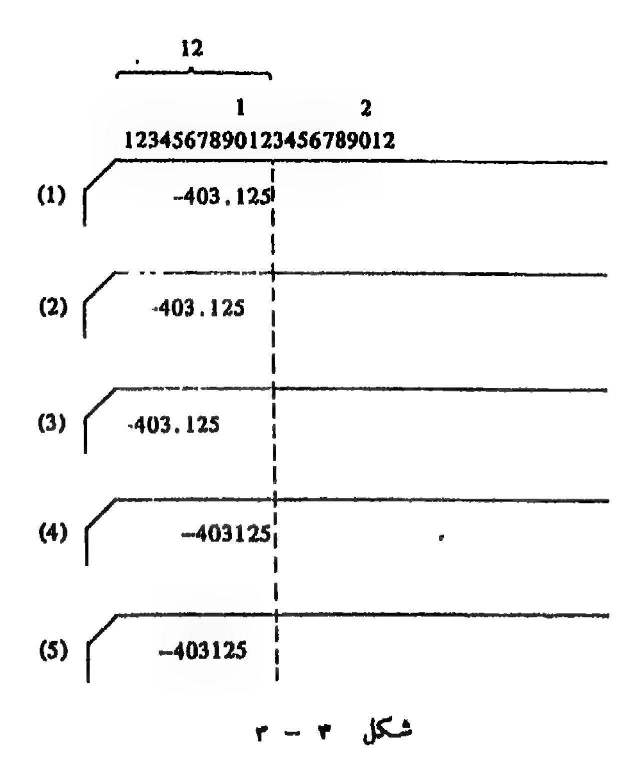
افرض أن المطلوب قراءة العدد 403.125 — ليخزن فى A باستخدام

```
      READ(5, 20) A
20    FORMAT(F12.4)
```

حيث أن عرض الحقل 12 ، فسيخصص الرقم الظاهر فى أول 12 عموداً من بطاقة البيانات إلى A . وعلى هذا ، يمكن أن تستخدم أيا من البطاقات الأولى فى شكل ٣ — ٣ كبطاقة للبيانات .

فى الشكل العشرى ، ليس من الضرورى تثقيب A مضبطة من الطرف الأيمن فى الحقل الخاص بها كما هو الحال فى الثوابت الرقمية الصحيحة . رغم أن المسافات الخالية فى الحقول الرقمية تفسر كأصفار ، إلا أن إضافة أصفار إلى رقم فى الشكل العشرى من الجهتين لا يغير من قيمه ، على سبيل المثال 403.125 — و 403.12500 — لهما نفس القيمة . لذا وكما ذكر سابقاً ، فعند قراءة حقل -F سيهمل الثابت *d* فى *Fw.d* طالما الرقم مثقب فى الشكل العشرى (أى يحتوى على علامة عشرية) ويقع بداخل الحقل المحدد . وسيقرأ هذا الرقم كما يظهر فى بطاقات البيانات . يجب أن نلاحظ أن عدد الأرقام المعنوية التى يحتفظ بها الحاسب هى خاصية تابعة للآلة .

لا تتطلب مترجمات كثيرة أن تثقب الثوابت الحقيقية الموصفة بحقل — F فى المدخلات بعلامة عشرية . فى هذه الحالة ، يكون الثابت الصحيح *d* فى *Fw.d* مؤثراً . وبالتحديد إن لم توجد علامة عشرية فى الحقل المحدد ، سيكون الرقم المقروء بالضبط عدد *d* من الأرقام العشرية ، وتضاف علامة عشرية بين العمود رقم *d* والعمود رقم (*d* + 1) بالعد من الطرف الأيمن من الحقل إلى اليسار . على سبيل المثال . افترض أننا استعملنا الزوج READ · FORMAT السابق لقراءة بطاقة 4 فى الشكل ٣ — ٣ . وحيث أنه لا توجد علامة عشرية بين الأعمدة 1 و 12 ، فأثناء القراءة بمواصفات الحقل F12.4 ستعتبر علامة عشرية بين العمود رقم 4 والعمود رقم 5 من الجانب الأيمن من الحقل ، أى بين العمود 9 و 8 .



شكل ٣ — ٣

بالتالى ، فيكون العدد المخصص إلى A هو 40.3125 – وبالمثل ، لو أننا استعملنا نفس الزوج READ-FORMAT لقراءة بطاقة 5 من الشكل ٣ – ١ لكان العدد المخصص لـ A هو 4031.25 — .

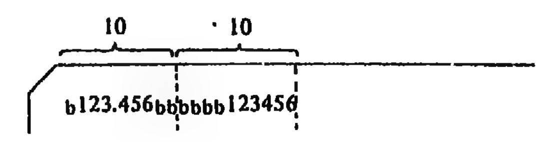
تهمل السمة السابقة الحاجه لتثقيب علامة عشرية فى الثوابت الحقيقية عند تحضير مجموعة بطاقات البيانات (واضح أن هذا سيوفر الوقت للتجهيزات التجارية) ومع ذلك عند استخدام هذه السمة فان مكان الرقم المثقب يؤثر على الرقم المخصص . وبالتالى ننصح مخططى البرامج المبتدئين بالتنازل عن هذه السمة ، أى الإلتزام بتثقيب العلامة العشرية عند استخدام حقول -F .

مثال ٣ – ٢

افرض أن زوج READ-FORMAT التالى :

```
   READ(5, 21) A, B
21 FORMAT(F10.2, F10.2)
```

قد استخدم لقراءة بطاقة مثقبة كالتالى :



حيث أن الرقم فى الأعمدة من 1 إلى 10 يحتوى على علامة عشرية ، فانه يخصص إلى A كما يظهر ، ومن ثم ستحتوى A على 123.456 وأن الرقم الثانى فى الأعمدة 11 إلى 20 لا يحتوى على علامة عشرية ، ومن ثم سيقرأ ويخصص إلى B 123.56

حقل E :

تذكر أولا ، أن الثابت الحقيقى فى الشكل الأسى (قسم ٢ – ٢) يتكون من جزءين الجزء الأول ثابت حقيقى فى الشكل العشرى ويتبع هذا الجزء الأسى (الجزء الثانى) ويبدأ بالحرف E يتبعه ثابت صحيح له إشارة أو بدون إشارة ولكن بحد أقصى عددان .

تقرأ الثوابت الحقيقية فى الشكل الأسى باستخدام حقل *E* . الشكل العام لمواصفات الحقل هو :

Ew.d

يدل الحرف *E* على أن البيانات حقيقية ، ويكتب الثابت فى الشكل الإسى ، تدل *w* وهى عدد صحيح بدون إشارة على عرض الحقل . وتدل *d* وهى عدد صحيح بدون إشارة على عدد الأرقام العشرية . كما فى حقل -F تهمل *d* فى المدخلات إذا تم تثقيب الرقم بعلامة عشريه .

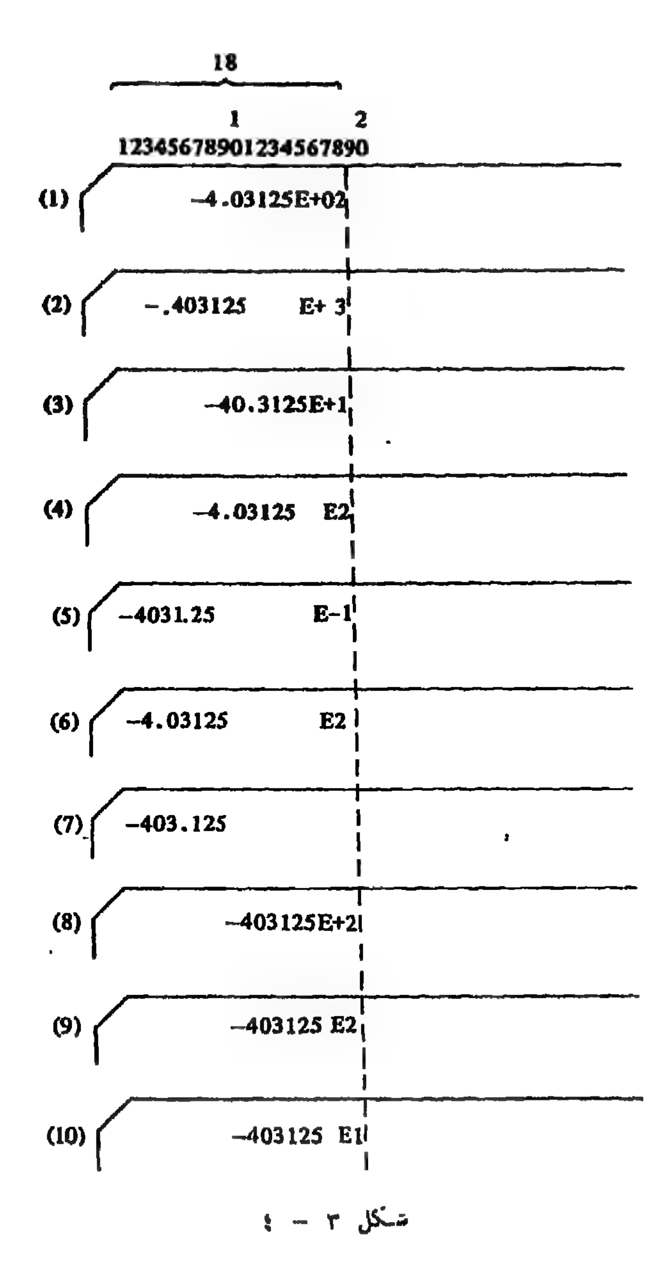
افرض أن المطلوب قراءة الرقم 4.03125E + 02 وتخزينه فى A باستخدام

```
   READ(5, 30) A
30 FORMAT(E18.5)
```

لاحظ أن عرض الحقل هو 18 وعلى هذا فسوف يخصص الرقم الظاهر فى أول 18 عموداً من بطاقة البيانات لـ A . وبذلك يمكن استخدام أى بطاقة من أول خمس بطاقات فى الشكل ٣ – ٤ كبطاقة بيانات لتعطى A القيمة $4.03125 \cdot 10^3$ — . إذا استخدمت البطاقة السادسة سوف يتم تخصيص الرقم 4.03125E + 20 — لـ A حيث تفسر المسافة الخالية فى العمود 18 كصفر وبالتالى من الضرورى أن يثقب الجزء الأسى مضبط من جهة اليمين فى حقل -E .

إذا كان ثابت حقيقى له إس صفر (أى 0.0 ، E) حينئذ يمكن عدم تثقيب جزء الأس . بمعنى آخر إذا تم تثقيب الرقم بغير الجزء الأسى ، تعامل مواصفات الحقل *Ew.d* بنفس طريقة *Fw.d* وبالتالى يمكن أن تستخدم البطاقة السابعة من الشكل ٣ – ٤ لتخصص

4.03125 E + 02 − إلى A (فى ضوء هذه المرونة ، يفضل بعض مخططى البرامج استخدام حقل E فى المدخلات بغض النظر إذا كانت بطاقات البيانات مثقبة بأس أم لا)



شكل ٣ − ٤

كما فى حقل -F فاستخدام العلامة العشرية فى المدخلات اختيارى عند استخدام حقل E . إن لم يكن هناك علامة عشرية فى الحقل المحدد (على يسار E) سنفترض علامة عشرية بين العمود *d* و *d* +1 مبتدأ بالعد من أول عمود على يسار الحرف E متجها إلى اليسار . على سبيل المثال ، افترض أننا استخدمنا الزوج السابق READ-FORMAT لقراءة بطاقة (8) من الشكل ٣ − ٤ . ولعدم وجود علامة عشرية فى الأعمدة 1 إلى 18 فسوف يتسبب العدد الصحيح 5 فى المواصفات E18.5 فى إضافة علامة عشرية بين العمود الخامس والسادس من يسار الحرف E أى بين العمود 10 و 11 . وبالتالى فلا يزال الرقم المخصص لـ A هو 4.03125E + 2 − . من ناحية أخرى ، لو استخدمنا نفس الزوج READ-FORMAT لقراءة البطاقة (9) من الشكل ٣ − ٤ فسيخصص

40.3125E2 – لـ A حيث تضاف العلامة العشرية بين العمود 11 و 12 (سيكون الحرف E هنا فى العمود 17 ؛ وبالمثل فسوف يختصر الرقم 40.3125E1 – لـ A إذا استخدمنا بطاقة (10) .

حقل X

ممكن أن نميز بين عدة ثوابت على نفس بطاقة البيانات وذلك بفصلهم بواسطة مسافة خالية أو أكثر . ويمكن تخطى هذه المسافات باستخدام حقل –X . وبصورة عامة ، مواصفات حقل –X تأخذ الشكل :

nX

n هى عدد صحيح بدون إشارة وتشير إلى عرض الحقل ، ويعنى الحرف X أنه يجب تخطى الحقل المناظر المتكون من عدد n من الأعمدة بغض النظر عما إذا كان الحقل يحتوى على معلومات أم لا .

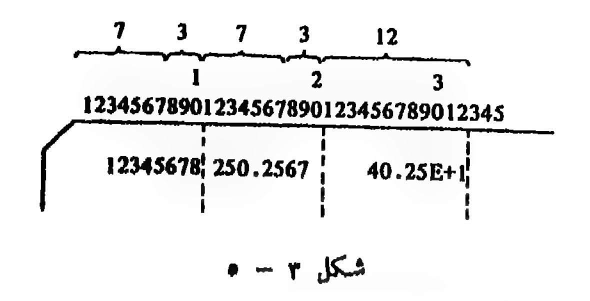
مثال ٣ – ٣

افترض أن برنامج به الجمل

```
      READ(5, 75) ID, AMOUNT, BALANS
   75 FORMAT(I7, 3X, F7.2, 3X, E12.2)
```

وافترض أنه قد تم قراءة بطاقة البيانات فى الشكل ٣ – ٥ . لاحظ أن عروض الحقول هى 7 ، 3 ، 7 ، 3 ، 12 على الترتيب . وعلى هذا ، سيخصص إلى ID العدد الصحيح فى أول حقل رقى (الأعمدة 1 إلى 7) وسوف تتخطى الأعمدة 8 إلى 10 . ثم يخصص الرقم الحقيقى فى ثانى حقل رقى إلى AMOUNT (الأعمدة 11 إلى 17) وسوف تتخطى الأعمدة 18 إلى 20 ؛ بعد ذلك يخصص الرقم الحقيقى فى ثالث حقل رقى إلى BALANS (الأعمدة 21 إلى 32) وعلى ذلك :

ID = 12345,   AMOUNT = 250.25,   BALANS = 402.5 = 40.25E+1



شكل ٣ – ٥

٣ – ٥ جمل WRITE المصاغة وتحكم العربة

يأمر الحاسب باخراج البيانات بواسطة جملة WRITE مصحوبة بجملة FORMAT وتأخذ الشكل التالى :

```
      WRITE(m, n) variable list
    n FORMAT(format list)
```

وبالتحديد فإن جملة WRITE تبدأ بكلمة WRITE . . يتبعها عدد صحيح m يحدد رقم وحدة الإخراج ورقم جملة n تحدد جملة FORMAT المصاحبة ، تفصل بين m و n فصلة ويحاطا بقوسين ويلى ذلك المتغيرات المطلوب طباعة قيمها مفصولة بفصلات . سنفترض أن آلة الطباعة هى التى تستخدم لطباعة مخرجاتنا . لذا فرقم وحدة الإخراج فى كل أمثلتنا سيكون m = 6 (انظر جدول ٣ – ١) .

يجب أن تكون جملة FORMAT المصاحبة معنونة برقم الجملة *n* وستأخذ نفس الشكل كما فى المدخلات . أى أنها تبدأ بكلمة FORMAT متبوعة بقائمة من مواصفات الحقول منفصلة عن بعضها بواسطة فصلات وتحاط بأقواس . وسنشرح معانى مواصفات الحقول فى الإخراج فى القسم التالى .

سيولد كل زوج من WRITE-FORMAT سيل من حروف الإخراج (سجل) . (المسافة الخالية هى حرف ويرمز إليها أحياناً بـ « b » ، أى b كدليل) . يتحكم أول حرف من سجل الإخراج فى حركة عربة آلة الطباعة تبعاً التعليمات فى جدول ٣ — ٢ ولكنه لا يطبع . بالتحديد يختبر الحاسب أول حرف من سجل الإخراج ثم يؤدى أمر التحكم فى العربة ، ثم يطبع بعد ذلك بقية الحروف فى سجل الإخراج .

جدول ٣ — ٢

الحروف الأولى	تعليمات التحكم	مدخل الصياغة
b (مسافة)	تقدم سطر واحد (فى العادة مسافة سطر واحد)	1X or' '
0 (صفر)	تقدم سطرين (يتخطى ، سطر الكتابة على مسافتين)	'0'
١	تقدم إلى أول الصفحة التالية	'1'
+	لا تتقدم	'+'

على سبيل المثال ، سجل الإخراج .

$_{bbbbb}12345_{bb}67.89$

يأمر الحاسب بالتقدم سطرا واحداً ثم يطبع بعد ذلك 4 مسافات خالية ثم 12345 ، ثم مسافتين خاليتين و 67.89 ، وسجل الإخراج

$0_{bbbb}12345_{bb}67.89$

يأمر الحاسب بالتقدم سطرين (يتخطى سطر) ثم يطبع 4 مسافات خالية ثم 12345 ، ثم مسافتين خاليتين ثم 67.89 ، وسجل الإخراج

$1_{bbbb}12345_{bb}67.89$

يأمر الحاسب بالتقدم إلى صفحة جديدة ثم يطبع 4 مسافات خالية ثم 12345 ثم مسافتين خاليتين ثم 67.89 .

تمد جملة FORMAT فى الزوج WRITE-FORMAT الحاسب بأنواع بيانات الإخراج ومكان ظهورها فى سجل الإخراج . بالتحديد يخصص لكل قيمة فى قائمة المتغيرات فى جملة WRITE مجموعة من المسافات (تسمى حقلا) . ذلك عن طريق مدخل مناظر فى جملة FORMAT . توضع دائماً القيمة العددية للمتغير فى الحقل الخاص بها .

مضبطة من الطرف الأيمن (right-justified) .

أى ينتهى الرقم فى آخر مسافة خالية من الحقل . وسوف نوضح ذلك بمثال رغم أننا سنناقش مداخل الصيغة بالتفصيل فى القسم التالى .

مثال ٣ — ٤

افترض أننا نريد طباعة القيم J ، K، L بالترتيب 128 = J ، 72 = K، 179 — = L يمكن إنجاز ذلك بزوج الجمل التالى .

```
      WRITE(6, 20) J, K, L
   20 FORMAT(I8, I6, I8)
```

يخبر الرقم 6 فى جملة WRITE الحاسب أن وحدة الإخراج هى آلة الطباعة ، ويدل الرقم 20 على أن القيم ستطبع طبقاً لجملة FORMAT التى تحمل الرقم 20 (يختار الرقم 20 عشوائياً) .

تحدد أول ثلاثة حقول فى سجل الإخراج بواسطة زوج WRITE-FORMAT الذى يحتوى الحروف التالية :

bbbbb128 (I8) تتولد القيمة الصحيحة لـ J مضبطة من الطرف الأيمن من أول حقل بعرض 8 ; أى أن قيمة J مزودة من اليسار بخمسة حروف خالية .

bbbb72 (I6) تتولد القيمة الصحيحة لـ K مضبطةمن الطرف الأيمن من الحقل التالى بعرض 6

bbbb—179 (I8) تتولد القيمة الصحيحة لـ L مضبطة من الطرف الأيمن من الحقل التالى بعرض 8 .

وبالتالى ، يحتوى سجل الإخراج على الحروف التالية :

سجل الإخراج bbbbb128bbbb72bbbb —179

وعلاوة على ذلك ، يخبر سجل الإخراج الحاسب بالتقدم سطراً واحداً ، ثم يطبع بعد ذلك 4 مسافات ثم 128 و 4 مسافات ، ثم 72 ، و4 مسافات ثم 179—. وتظهر المخرجات المطبوعة فى الشكل ٣ — ٦ . صفحات الطباعة لها عروض مختلفة ، وصفحة الطباعة فى الشكل ٣ — ٦ تفترض أن هناك 132 عمودا للطباعة .

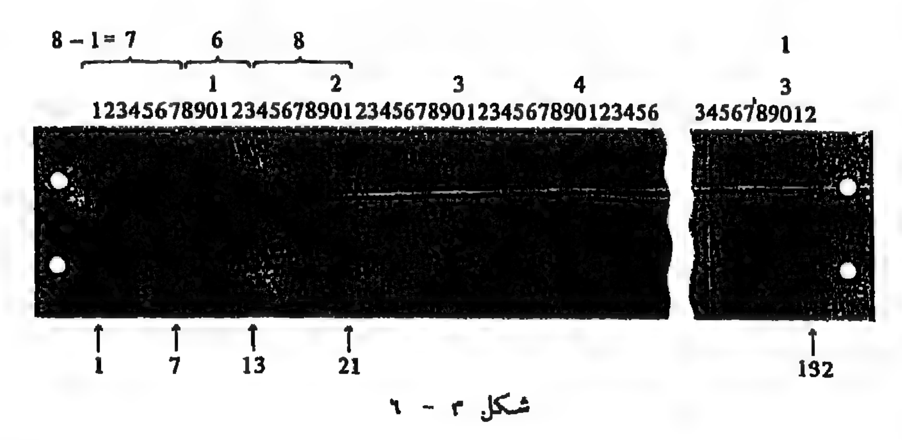
تلخيصاً لما سبق يؤدى الحاسب الخطوات التالية فى أمر WRITE-FORMAT :

١ — يولد سجل الإخراج المنشأ بزوج WRITE-FORMAT

٢ — يختبر أول حرف وينفذ أمر التحكم فى العربة .

٣ — يطبع بقية الحروف من سجل الإخراج

نلاحظ أنه بعد إتمام أمر WRITE تتوقف آلة الطباعة دائماً على السطر الذى تمت طباعته . ومن ثم ، فأول حرف مولد بواسطة جملة إخراج تالية سيستخدم مرة أخرى التحكم فى العربة .



شكل ٣ — ٦

ملحوظة : من العادات الجيدة فى البرمجة أن يعطى الفرد عادة إشارة صريحة للتحكم فى العربة . يتم هذا باختيار أحد المداخل فى جدول ٣ – ٢ كمدخل أول فى جملة FORMAT . سنناقش ونشرح هذا فى القسم التالى (بهذه العادة يمكن تجنب الأخطاء الموضحة فى المسائل ٣ – ١٢ (ب) ، ٣ – ٣٢ (و) ، و ٣ – ٣٣ (و)) . لاحظ أن المثال السابق لا يحتوى على إشارة صريحة للتحكم فى العربة ، ولكن إشارة التحكم فى العربة كانت متضمنة فى مواصفات الحقل I8 التى أعطت حقلاً أكبر من القيمة J .

٣ – ٦ مواصفات حقل الاخراج

كما نوقش من قبل ، فإن جملة FORMAT المصاحبة لجملة WRITE تحدد أنواع البيانات وأماكن المعلومات التى ستطبع . فى هذا القسم سوف نغطى نفس أنواع مواصفات الحقل التى غطيت فى الإدخال .

wX, *Iw*, *Fw.d*, *Ew.d*

ويلزم التأكد بأن معانى هذه المواصفات فى الإخراج قد تختلف عنها فى الإدخال . والسبب الأول فى ذلك أن كل الثوابت الرقمية تطبع مضبطة من الطرف الأيمن فى الحقول الخاصة بها ومضاف إليها على اليسار مسافات خالية . كذلك ، تلعب *d* فى *Fw.d* و *Ew.d* دوراً هاماً فى الإخراج ، فى حين أن *d* قد تلغى أحياناً فى المدخلات .

حقل X

يمكن أن نحصل على مسافات خالية الفصل بين عدة ثوابت على نفس السطر باستعمال حقل –X . الشكل العام لمواصفات الحقل هو :

wX

حيث تدل *w* على عرض الحقل . وتأثير *wX* هو توليد عدد *w* من المسافات الخالية فى سجل الإخراج .

وعلى وجه الخصوص ، إذا ظهرت 1X كأول مدخل فى جمل FORMAT حينئذ تتولد مسافة كأول حرف فى سجل الإخراج . تستخدم هذه المسافة للتحكم فى العربة فتجعلها تتقدم سطراً واحداً . بمعنى آخر ، إذا كانت جملة FORMAT فى الشكل

```
n   FORMAT(1X,...)
```

فإنها تؤدى دائماً إلى مسافة واحدة بين السطور ، بالمثل .

```
n   FORMAT(10X,...)
```

تعنى دائماً مسافة واحدة بين السطور مع الزحزحة تسعة مسافات للداخل .

حقل — I

تطبع الثوابت الصحيحة باستعمال حقل — I . الشكل العام لمواصفات الحقل هو :

Iw

حيث تدل *I* على النوع الصحيح و *w* عرض الحقل .

على سبيل المثال ، افرض أن M تحتوى 250 و N تحتوى 46 وتم تنفيذ

```
    WRITE(6, 10) M, N
10  FORMAT(1X, I6, 3X, I4)
```

يولد المدخل 1X مسافة واحدة b ، يولد المدخل I6 الذى يناظر M 250 bbb أى أن قيمة M تتولد مضبطة من الطرف الأيمن من الحقل التالى ذى العرض 6 . يولد المدخل 3X ثلاث مسافات bbb يولد المدخل I4 الذى يناظر N 46 bb أى أن قيمة N تتولد مضبطة من الطرف الأيمن من الحقل التالى ذى العرض 4 وعلى هذا ، يتولد سجل الإخراج التالى .

bbbb250bbbbb46 . سجل الإخراج

وحيث أن أول حرف من سجل الإخراج هو مسافة خالية وذلك يعنى تباعداً مفرداً للسطور ، فستتقدم آلة الطباعة سطراً واحداً وتطبع بقية الحروف :

bbb250bbbbb46 : الحروف المطبوعة

إذا لم يتوافر عرضاً كافياً ، أى إذا احتوت قيمة رقمية على حروف أكثر من الحروف الموصوفة فى عرض الحقل ، ستبتر بعض المترجمات القيمة من اليسار (أو اليمين) وتطبع بقية الحروف . فى المثال السابق ، لو احتوت M على 54321 إذن ستطبع :

bbb250bbb4321

ولا تعطى أى رسالة لتشير إلى ذلك . لحسن الحظ ، فان معظم المترجمات المبينة بطريقة مناسبة وستشير إلى نقص عرض الحقل بطبع نجوم فى الحقل المعين . باستعمال المثال السابق مرة أخرى مع تحديد M = 54321 فسوف يطبع

bbb250bbb****

يجب أن يراعى مخططو البرامج أن يكون عرض الحقل كبيراً بدرجة كافية لتلائم كل قيم المتغيرات الممكنة .

حقل -F :

يمكن أن تطبع الثوابت الحقيقية فى الشكل العشرى باستعمال حقل -F . الشكل العام لمواصفات الحقل هو :

Fw.d

حيث تشير *F* إلى نوع البيانات و *w* إلى عرض الحقل . يشير العدد الصحيح *d* إلى عدد الأرقام العشرية ويتم هذا دائماً بالتقريب . بمعنى آخر ، تخبر *Fw.d* الحاسب بأن يقرب الرقم الحقيقى المناظر إلى *d* رقماً عشرياً ، وأن يطبع الرقم مضبطاً من الطرف الأيمن فى الحقل الخاص به ذى العرض *w* .

مثال ٣ — ٥

افرض ID = 125 و AMOUNT = 450.2462 وتم تنفيذ

```
      WRITE(6, 27) ID, AMOUNT
27    FORMAT(I8, 3X, F10.2)
```

لاحظ أن مواصفات الحقول تحتوى على عرض الحقول 8 ، 3 ، 10 على الترتيب . تخبر مواصفات الحقل I8 والتى تناظر ID الحاسب بأن يولد القيمة ID مضبطة من الطرف الأيمن فى أول حقل ذى العرض 8 من سجل الإخراج ؛ لذا تتولد bbbbb125 تولد 3X ثلاث مسافات خالية : bbb . تخبر F10.2 الحاسب أن يقرب قيمة A إلى رقمين عشريين أى إلى 450.25 وأن يولد هذه القيمة مضبطة من الطرف الأيمن فى الحقل التالى ذى العرض 10 لذا تتولد bbbb450.25 وبذلك يصبح سجل الإخراج الذى تم توليده :

bbbbb125bbbbbbb450.25 : سجل الإخراج

حيث يستخدم أول حرف للتحكم فى العربة ، سيظهر الخرج على صفحة الطباعة كما فى الشكل ٣ — ٨ .

وأننا لنتوكد أن الرقم 450.25 يشغل ست مسافات فى الخرج حيث أن العلامة العشرية تشغل مسافة طباعة واحدة .



شكل ٣ — ٧

كما فى حقل ـI إن لم يتح عرض كاف ، أى ، إذا كانت القيمة العددية فى حقل ـF تحتوى حروفاً أكثر من عرض الحقل ، قد يبتر الرقم من اليسار (أو اليمين) أو قد تطبع نجمة . وعلى هذا ، يجب أن نتوخى الحرص عند استعمال حقل ـF وأن يكون عرض الحقل كبيراً بدرجة كافية ليلائم الرقم بالتحديد ، ان كان العدد المطلوب من الأرقام العشرية هو d و n هى عدد خانات الجزء الصحيح ، فعلى الأقل يجب أن يكون $w \geq n + d + 2$. تسمح 2 بمسافة للاشارة ومسافة العلامة العشرية . على سبيل المثال إذا كانت $|X| < 1000$ ونحن نحتاج إلى رقمين عشريين فقط ، إذن تكون F7.2 كافية : $w = 3 + 2 + 2$ حيث تتاح ثلاث مسافات للجزء الصحيح من X .

يجب أن نتوخى الحرص حيث عدد الأرقام العشرية كبيراً بدرجة كافية . على سبيل المثال ، افرض أن A تحتوى على 0.246E—05 ومواصفات الحقل هى F12.4 (تخزن الأرقام الحقيقية فى الحاسب بالشكل العلمى ، أنظر ملحق أ) . حينئذ يطبع .

0.0000

فقط حيث أن قيمة A ، وهى 0.00000246 فى الشكل العشرى ستكون 0.0000 عندما تقرب إلى أربعة أرقام عشرية .

حقل E

يمكن أن نطبع الثوابت الحقيقية فى الشكل الأسى باستعمال حقل ـE . الشكل العام لمواصفات الحقل هو

Ew.d

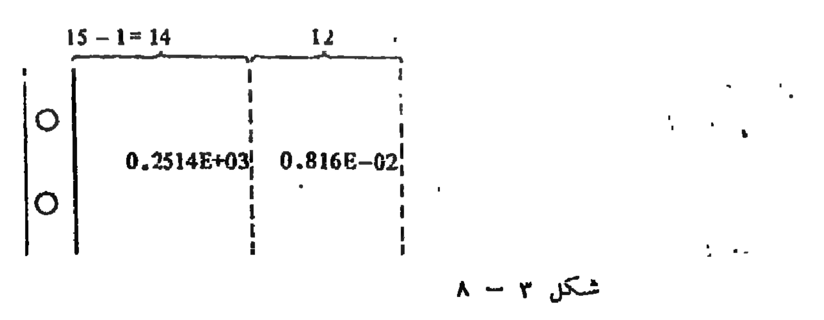
حيث تشير E إلى نوع البيانات وتشير w إلى عرض الحقل . يشير العدد الصحيح d إلى عدد الأرقام المعنوية ويتم هذا بالتقريب . بالإضافة إلى ذلك ، تطبع الثوابت الحقيقية فى الخرج بالشكل الأسى المعيارى ، أى كثوابت بين 0.1 إلى 1.0 (0.1 — إلى 1.0 —) بصحبة الأس المناسب الذى يحتوى دائماً على أربعة حروف E ± XX بمعنى آخر تخبر *Ew.d* الحاسب أن يقرب الرقم الحقيقى المناظر إلى العدد d من الأرقام معنوية وأن يطبع العدد فى الشكل الأسى المعيارى مضبطاً من الطرف الأيمن فى الحقل الخاص به ذى العرض w . (ونلاحظ أن فى الشكل الأسى المعيارى ، عدد الأرقام المعنوية مساوى أيضاً لعدد الأرقام العشرية)

مثال ٣ — ٦

افرض $A = 251.381$ و $B = 0.0081626$ وافرض أنه تم تنفيذ

```
   WRITE(6, 22) A, B
22 FORMAT(E15.4, E12.3)
```

وحيث أن الأرقام الحقيقية (فى الشكل E أو F) تخزن داخلياً فى الشكل الأسى المعيارى (أنظر ملحق أ) . تحتوى A على 0.251381 E+03 + وتحتوى B على 02 — 0.81626 E + . يخبر المدخل E 15.4 الحاسب أن يقرب قيمة A إلى أربعة أرقام معنوية أى 03 + 0 2514E ويضعها مضبطة من الطرف الأيمن فى الـ15 مسافة الأولى من سجل الإخراج . ويخبر المدخل E12.3 الحاسب أن يقرب قيمة B إلى ثلاثة أرقام معنوية أى ، 02 — 0.816E ويضعها مضبطة من الطرف الأيمن فى 12 مسافة التالية من سجل الإخراج . وبالتالى فإنهما تظهر كما هو موضح فى الشكل ٣ — ٨ .

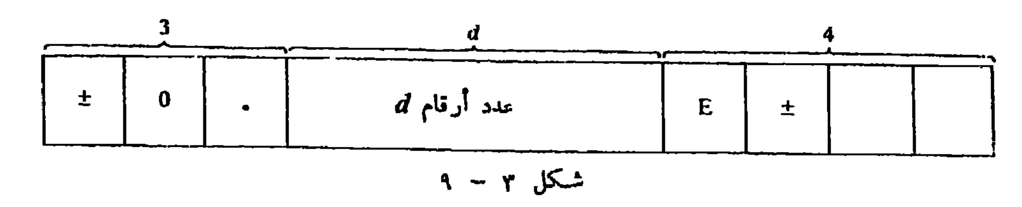


شكل ٣ — ٨

لإتاحة عرض كاف لطبع الأرقام العشرية فى شكل -E فإننا نحسب w كما يلى : إذا كان المطلوب عدد d من الخانات المعنوية يجب أن يحقق عرض الحقل w الشرط التالى :

$$w \geq d + 7$$

ويمكن الحصول عليه كالآتى : أربع مسافات مطلوبة للأس E ± XX ، مسافة واحدة للإشارة ومسافة واحدة الصفر الموجود على اليسار ، مسافة واحدة العلامة العشرية . ويبين شكل ٣ — ٩ أيضاً طريقة الحساب .



شكل ٣ — ٩

نلخص قسمى الإدخال والإخراج بالتعليقات التالية :

١ — يمكن أن تحتوى جملة FORMAT على كثير من مواصفات الحقول . ويجب أن تكون الحقول متلاحقة ، أى تلى بعضها الآخر .

٢ — لقراءة قيم أو طباعتها ، يجب أن يكون هناك تناظر فى ترتيب وضع المتغيرات مع ترتيب مواصفات الحقول الرقمية (أو الحرفية) فى جملة FORMAT المناظرة ويجب أن تتفق أيضاً فى النوع .

٣ — يجب أن تصاحب كل جملة READ أو WRITE جملة FORMAT ومع ذلك يمكن لجملة FORMAT واحدة أن تصاحب (يشار إليها من) عدة جمل I/O . وعلاوة على ذلك ، فان جمل FORMAT جمل غير منفذة .

٣ ــ ٧ الحقل الحرفى

افترض أننا نرغب فى طباعة عنوان POPULATION OF USA. فلإنجاز ذلك نحيط الرسالة بفصلات عليا داخل جملة FORMAT كالآتى :

```
      WRITE(6, 30)
   30 FORMAT(1X, 'POPULATION OF USA')
```

عند تنفيذ جملة WRITE السابقة ، سيتم توليد كل الحروف بداخل الفصلات العليا . ويحتوى سجل الإخراج :

سجل الإخراج : ꞗPOPULATIONꞗOFꞗUSA

حيث تولد المسافة الأولى بسبب 1X . آخذين فى الاعتبار التحكم فى العربة ، بذلك تطبع الرسالة POPULATION OF USA

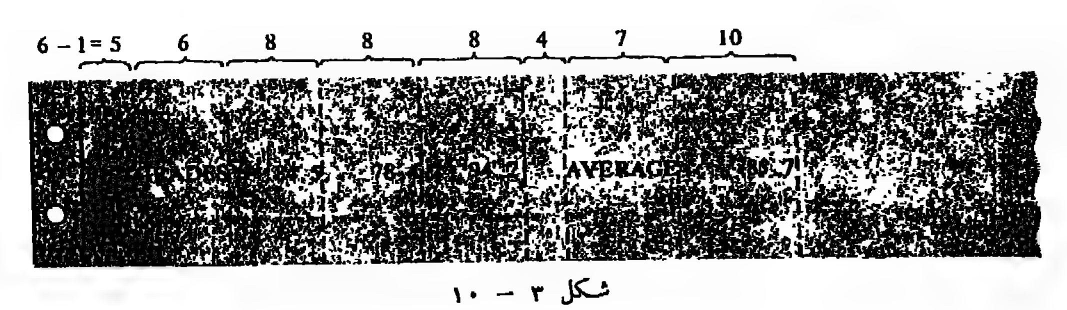
يمكن أن تطبع رسالة حرفية مع قيم رقمية . افرض G1 = 84.5 ، G2 = 78.4 ، G3 = 94.2 ، AVE = 85.7 . فسوف تولد التعليمات :

```
      WRITE(6, 80) G1, G2, G3, AVE
 80   FORMAT(6X, 'GRADES', 3F8.1, 4X, 'AVERAGE', F10.1)
```

سجل الإخراج التالى :

سجل الإخراج : ꞗꞗꞗꞗꞗꞗGRADESꞗꞗꞗꞗ84.5ꞗꞗꞗꞗ78.4ꞗꞗꞗꞗ94.2ꞗꞗꞗꞗAVERAGEꞗꞗꞗꞗꞗꞗ85.7

سوف يظهر الخرج على ورقة الطباعة كما فى الشكل ٣ — ١٠ . لاحظ أن المدخل 6X سيتسبب فى طباعة 5 = 1 — 6 مسافات خالية فقط ، حيث تستخدم أول مسافة للتحكم فى العربة . لاحظ أيضاً أن الرسائل GRADES و AVERAGE تشغل حقولا ذات عروض 6 و 7 على الترتيب ، وهى عدد الحروف فى الرسالة .



شكل ٣ — ١٠

(لاحظ أن 3F8.1 هى اختصار F8.1,F8.1,F8.1 سوف نذكر ذلك فى قسم ٣ — ٩) .

أحد الاستخدامات الأساسية لحقل الحرفى هو للتحكم فى العربة . افرض أننا أردنا طباعة الرسالة POPULATION OF USA فى بداية صفحة جديدة . يمكن إنجاز ذلك بواسطة الجمل :

```
      WRITE(6, 20)
 20   FORMAT('1', 'POPULATION OF USA')
```

ويكون أول مدخل للصيغة 1 هو أمر التحكم فى العربة ليذهب إلى صفحة جديدة . بالمثل . يمكن أن تستخدم '0' كأول مدخل الصيغة إذا أردنا أن يتخطى الحاسب سطراً قبل طباعة POPULATION OF USA أو سوف نستخدم + كأول مدخل الصيغة . إذا أردنا أن يطبع الحاسب على السطر الحالى . توجد قائمة بهذه المداخل فى أماكنها المناسبة فى العمود الثالث من جدول ٣ — ٢ الذى يحتوى على تعليمات التحكم فى العربة .

٣ — ٨ السجلات ، السجلات المتعددة ، الشرطة المائلة (/)

السجل هو مجموعة البيانات المناظرة المتغيرات فى قائمة (I/O) . والحد الأقصى لطول السجل محدود بوحدة (I/O) التى ذكرناها فى هذا الشرح وهى : 80 حرف للبطاقات ، 72 حرف للآلة الكاتبة المركزية ، 132 حرف لوحدات الطباعة . بالطبع ، فان الطول الحقيقى يستمد من أمر (I/O) المعطى فى جملة FORMAT .

حتى الآن فإن كل أمر من أوامر (I/O) نقل سجل واحد من المعلومات . يقرأ زوج الجمل READ-FORMAT من بطاقة إدخال واحدة ويطبع كل زوج جمل WRITE-FORMAT على سطر خرج واحد . إلا أنه ، يمكن أن تستخدم جملة FORMAT واحدة لتعريف سجلات متعدده . وسنناقش فيما يلي طريقتان من هذه الطرق :

الحالة (أ)

تذكر أن تنفيذ جملة READ/WRITE لا تتم إلا إذا كانت كل المتغيرات المذكورة قد أعطيت قيم أو طبعت . افترض أنه تم تنفيذ

```
      READ(5, 20) A, B, C, D
   20 FORMAT(F8.2, 2X, F15.7)
```

لاحظ أن عدد المتغيرات (4) وذلك أكثر من المواصفات الرقية (2) . وبذلك فإننا نقرأ المتغيرين A و B من بطاقة 1 أولا من الأعمدة 1 إلى 8 و 11 إلى 25 بالمواصفات F8.2 و F15.7 على الترتيب . وحيث أن هناك متغيرات أخرى يجب قراءتها ، فسوف تستخدم نفس جملة FORMAT مرة ثانية ، ولكن مبتدأة بسجل جديد ، أى بطاقة أخرى . ومن ثم ، تقرأ قيم C و D من بطاقة 2 من الأعمدة 1 إلى 8 و 11 إلى 25 على الترتيب (أنظر أيضاً قسم ١٠ - ٦) .

بالمثل ، سوف تنتج الجملتين

```
      WRITE(6, 30) A, B, C, D
   30 FORMAT(5X, F8.2, 2X, E14.7)
```

سجلين للمخرجات .

$_{bbbbb}$ + XXXX.XX$_{bb}$ + 0.XXXXXXXE + XX
$_{bbbbb}$ + XXXX.XX$_{bb}$ + 0.XXXXXXXE + XX

ومن ثم ، ستطبع A و B على سطر واحد تبعاً لـ F8.2 و E14.7 على الترتيب مع مراعاة ترك 4 مسافات إلى الداخل . وستطبع C و D على السطر التالى بنفس الطريقة .

ملحوظة : فى الحالة العكسية حيث توجد هناك مواصفات حقول أكثر من المتغيرات . فيتم تنفيذ جملة READ/WRITE بمجرد أن تقرأ / تطبع كل المتغيرات وبالتالى تهمل بقية المواصفات الزائدة . على سبيل المثال إذا غيرنا جملة READ السابقة إلى :

```
      READ(5, 20) A, B, C, D, E
```

وبذلك تقرأ ثلاث بطاقات وسيخصص الرقم الموجود فى الأعمدة 1 إلى 8 من البطاقة الثالثة إلى E باستعمال F8.2 .

الحالة (ب) :

يمكن أيضاً أن نستخدم الشرطات المائلة (/) لتعريف السجلات المتعددة . تستخدم الشرطات المائلة (/) فى جملة FORMAT للاشارة إلى انهاء السجل ، وعلى هذا ، تقرأ الجملتان التاليتان سجلين :

```
      READ(5, 40) A, B, C, D
   40 FORMAT(F10.2, F10.2/E15.7, E15.7)
```

أى تقرأ A و B من بطاقة 1 باستعمال F10.2 وتقرأ C و D من بطاقة 2 باستعمال E15.7 . بالمثل فان الجملتين :

```
      WRITE(6, 50) N, AVE
   50 FORMAT('1', 'NUMBER OF STUDENTS =', I3/1X, 'EXAM AVERAGE =', F7.2)
```

تطبعان سجلين وسيطبع فى نهاية صفحة جديدة السطرين التاليين :

```
NUMBER OF STUDENTS = XXX
EXAM AVERAGE = XXX.XX
```

ونؤكد أنه يجب أن نذكر حرف التحكم فى العربة لكل سجل خرج جديد . وبالتحديد فان 1X التى تلى الشرطة المائلة فى جملة FORMAT السابقة هى أمر التحكم فى العربة للسجل الجديد .

يمكن أيضاً أن تظهر عدة شرطات مائلة متتالية – إلا أن ، الشرطات المائلة الظاهرة فى آخر جملة FORMAT تعطى نتيجة مختلفة اختلافاً بسيطاً عن التى تظهر فى سجل الإدخال وبالتحديد :

١ – فى الإخراج . عند ظهور عدد n من الشرطات المائلة المتتالية ، فسوف تطبع عدد $n - 1$ من الأسطر الخالية وعلى سبيل المثال كلا من

```
   WRITE(6, 10) J, K                   WRITE(6, 20) J, K
10 FORMAT(1X, I10///1X, I10)    و   20 FORMAT(1X, I10///)
```

سينتج سطران خاليان بين J و K .

٢ – فى الإدخال عند ظهور عدد n من الشرطات المائلة المتتالية فى الوسط فسوف تتخطى عدد $n - 1$ من البطاقات ، ولكن ، عند ظهور عدد n من الشرطات المائلة المتتالية فى النهاية فسوف يتخطى عدد n من البطاقات وعلى سبيل المثال فسوف يتسبب :

```
   READ(6, 30) J, K
30 FORMAT(I10///I10)
```

فى قراءة J من البطاقة الأولى وقراءة K من البطاقة الرابعة أى أنه سوف يتخطى بطاقتين . بينما سوف يتسبب :

```
   READ(6, 40) J, K
40 FORMAT(I10///)
```

فى قراءة J من البطاقة الأولى وقراءة K من البطاقة الخامسة أى أنه سوف يتخطى ثلاث كروت (أنظر المسائل ٣ – ٣٩ و ٣ – ٤٠ أيضاً) .

٣ – ٩ معامل التكرار

إفرض أن المطلوب قراءة / كتابة أربعة أرقام A و B و C و D باستعمال نفس المواصفات F10.2 فيجب أن تكون جملة FORMAT هى

```
FORMAT(F10.2, F10.2, F10.2, F10.2)
```

ولتبسيط ذلك يمكن أن نستخدم معامل التكرار 4 :

```
FORMAT(4F10.2)
```

وبذلك تكون جملتا FORMAT لهما نفس المعنى تماماً . بالمثل الجملتين التاليتين :

```
FORMAT(2I4, 3E15.7)
```

و

```
FORMAT(I4, I4, E15.7, E15.7, E15.7)
```

لهما نفس المعنى .

يمكن أيضاً أن نكرر مجموعة من مواصفات الحقول وعلى سبيل المثال :

```
FORMAT(5X, 2(I5, 3X, F8.2, 2X))
```

تعنى :

```
FORMAT(5X, I5, 3X, F8.2, 2X, I5, 3X, F8.2, 2X)
```

والجملة :

```
FORMAT(3(3X, I4), 2(5X, E9.2))
```

هى نفس الشيء مثل .

```
FORMAT(3X, I4, 3X, I4, 3X, I4, 5X, E9.2, 5X, E9.2)
```

وبهذه الاختصارات تكون جملة FORMAT أكثر إيجازاً وأسهل كتابة .

٣ — ١٠ برنامج بسيط كامل

لقد غطينا مادة كافية تمكننا من كتابة بعض البرامج الكاملة البسيطة . أحد هذه البرامج مكتوب هنا والبعض الآخر معطى فى قسم المسائل المحلولة .

افترض أن أطوال أضلاع مثلث T هى a و b و c والمحيط P للمثلث هو مجموع هذه الأطوال :

$$P = a + b + c$$

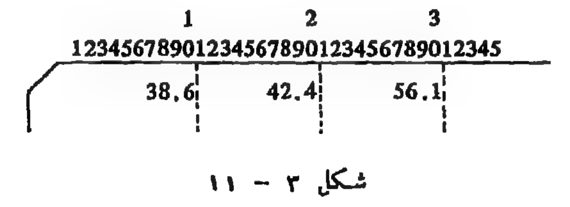
تعطى المساحة T بالمعادلة :

$$\text{Area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = P/2 = (a + b + c)/2.$$

حيث

افرض أننا نريد أن نحسب محيط ومساحة مثلث أطوال أضلاعه 38.6 و 42.4 و 56.1 سوف نثقب أولا هذه الأرقام على بطاقة بيانات باستعمال ، حقول بعرض 10 مثلا ، كما فى شكل ٣ — ١١ .



شكل ٣ — ١١

برنامج الفورتران الذى يجرى الحسابات السابقة هو :

```
C
C      PROGRAM CALCULATING PERIMETER AND AREA OF A TRIANGLE
C
       READ(5, 11) A, B, C
  11   FORMAT(3F10.1)
       P = A + B + C
       S = P/2.0
       AREA = SQRT(S*(S - A)*(S - B)*(S - C))
       WRITE(6, 12) A, B, C, P, AREA
  12   FORMAT(3(F8.1, 4X), F10.1, 4X, F12.2)
       STOP
       END
```

شرح وتنفيذ البرنامج

الأسطر ١ – ٣ هذه الأسطر تعليق حيث أن C مثقبة فى أول عمود .

الأسطر ٤ – ٥ تخبر هذه الأسطر الحاسب أن يقرأ قيم A و B و C من بطاقة البيانات باستعمال حقول كل منها بعرض 10 . ومن ثم ، تخصص القيم الآتية إلى A و B و C :

$A \leftarrow 38.6, \quad B \leftarrow 42.4, \quad C \leftarrow 56.1$

سطر ٦ يجمع الحاسب قيم A و B و C وتخصص هذه القيمة (137.1) للمتغير P :

$P \leftarrow 137.1$

سطر ٧ يخصص الحاسب P/2 للمتغير S :

$S \leftarrow 68.55$

سطر ٨ يوجد الحاسب قيمة التعبير الحسابى على اليمين ، حيث SQRT تعنى الجذر التربيعى ، ويخصص هذه القيمة 817.56570 إلى AREA :

$AREA \leftarrow 817.56570$

الأسطر ٩ – ١٠ يطبع الحاسب القيم A و B و C و P و AREA باستعمال حقول ذات عرض 7 = 1 — 8 ، 8 ، 8 ، 8 ، 10 ، 12 على الترتيب كما فى شكل ٣ – ٢ . لاحظ أن هذه الحقول تفصلها حقول خالية بعرض أربعة حروف (4X) لاحظ أيضاً أن P تم تقريبها إلى رقم عشرى واحد (F10.1) وتم تقريب AREA إلى رقمين عشريين (F12.2) .

سطر ١١ يخبر الحاسب بأن يتوقف STOP .

سطر ١٢ تخبر جملة END المترجم بعدم وجود جمل أخرى فى البرنامج تتطلب الترجمة ولن تكون فى البرنامج وقت التنفيذ .

8 – 1 = 7 4 8 4 8 4 10 4 12

1 2 3 4 5 6 7

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345

شكل ٣ – ١٢

ملحوظة : عند تنفيذ البرنامج السابق ، أشرنا إلى أن 38.6 خصصت إلى A . والرقم الحقيقى الذى تم تخزينه فى A هو 0.38600000E+02 . حيث أن التمثيل الداخلى للقيم الحقيقية تكون فى الشكل الأسى بأرقام عشرية تتراوح ما بين 8 إلى 9 تقريباً (أنظر ملحق أ) .

مسائل محلولة

إدخال / إخراج غير مساغ

٣ - ١ اكتشف الأخطاء إن وجدت ، فى جمل I/O التالية غير المصاغة

(أ) READ A. B. X. Y
(ب) PRINT NUMB, INT. RATE
(ﺟ) READ, ID, WAGE, MAX, MIN
(د) PRINT, ROOT, COEF, RANGE

(أ) يجب أن تكون هناك فصلة بعد READ .

(ب) يجب أن تكون هناك فصلة بعد PRINT ويجب ألا توجد فصلة بعد RATE .

(ﺟ) صواب

(د) يجب ألا توجد فصلة بعد RANGE .

٣ - ٢ افترض تثقيب ثلاث بطاقات بيانات كالآتى :

أول بطاقة : 22.2,3.33,444

ثانى بطاقة : 555,666,77.77

ثالث بطاقة : 8.888,9.99

أوجد القيم المخصصة للمتغيرات عند تنفيذ الآتى :

(أ) READ A, B
READ, J, K, X, Y

(ب) READ, A, B, J, K
READ, X, Y

(أ) عندما ينفذ الحاسب أول جملة READ يخصص 22.2 إلى A و 33.3 إلى B وبذلك يتم تنفيذ أول جملة READ وتهمل البطاقة الأولى رغم أن القيمة الثالثة لم تقرأ بعد . وعندما ينفذ الحاسب جملة READ الثانية ، سيقرأ من البطاقة الموجودة على قمة مجموعة البطاقات ، بطاقة البيانات الثانية . ومن ثم سيخصص 555 إلى J ، 666 إلى K ، 77.77 إلى X ثم يقرأ البطاقة التالية ليخصص 8.888 إلى Y وبذلك ، نحصل على :

A = 22.2, B = 3.33, J = 555, K = 666, X = 77.77, Y = 8.888

(ب) تعطى جملة READ الأولى :

A = 22.2, B = 3.33, J = 444, K = 555

وتعطى جملة READ الثانية :

X = 8.888, Y = 9.99

نلاحظ أنه تم إهمال بطاقة البيانات الثانية بعد تنفيذ جملة READ الأولى . وبذلك استخدمت بطاقة البيانات الثالثة لتنفيذ جملة READ الثانية .

٣ – ٣ افرض أن المطلوب تخصيص القيم 123 ، 456 ، 7.77 ، 8.88 و 9.99 للمتغيرات L ، M ، X ، Y و Z على الترتيب . اكتب جملة READ غير مصاغة ووضح كيف يجب أن تثقب البيانات على بطاقات البيانات إذا : (أ) استخدمت بطاقة بيانات واحدة فقط ، (ب) استخدمت بطاقتان البيانات القيم الصحيحة على البطاقة الأولى والقيم الحقيقية على البطاقة الثانية .

فى كلا الحالتين لدينا جملة READ التالية :

```
READ, L, M, X, Y, Z
```

يجب أن تثقب البيانات كالتالى :

(أ) أول بطاقة : 123, 456, 7.77, 8.88, 9.99

(ب) أول بطاقة : 123, 456

ثانى بطاقة : 7.77, 8.88, 9.99

٣ – ٤ اعتبر أنه تم تثقيب بطاقة بيانات كالتالى :

بطاقة بيانات : 123, 44.4

(أ) حدد المخرج عند تنفيذ جزء البرنامج التالى :

```
READ, J, X
K = J**2
Y = 3.0*X
PRINT, J, K, Y
```

(ب) اشرح الفرق إذا استبدلت جملة PRINT الوحيدة السابقة بالجملتين التاليتين

```
PRINT, J, K
PRINT, Y
```

(أ) تنفيذ البرنامج :

سطر ١ تخصص قيم بطاقة البيانات إلى J و X وينتج عن ذلك أن J = 123 ، X = 44.4

سطر ٢ تخصص مربع J إلى K وبهذا تكون قيمة K ← 15129 .

سطر ٣ يخصص ثلاثة أضعاف قيمة X إلى Y وبهذا تكون قيمة Y = 133.2

سطر ٤ تطبع قيم J و K و Y مع قيمة Y فى الشكل الأسى القياسى . وبذلك يظهر المخرج كالآتى :

```
123     15129     0.1332E 03
```

(ب) ستظهر قيم J و K على سطر واحد وقيمة Y على السطر الثانى وبذلك يظهر المخرج كالتالى :

```
       123     15129
0.1332E 03
```

الإدخال المصاغ

٣ — ٥ حدد موقع الخطأ ، إن وجد ، فى كل جملة من جمل READ وجملة FORMAT المصاحبة لها .

(أ)
```
   READ(5, 11) A, K, M, Z,
11 FORMAT(F8.0, I15, I10, I15)
```

(ب)
```
   READ(5, 11) A, B, J, K, L
11 FORMAT(3F8.1, 2I8)
```

(جـ)
```
   READ(5, 11), A, B, C, D, J, K, L
11 FORMAT(4F15.2, 3I15)
```

(أ) أولا ، لا يجب أن تكون هناك فصلة بعد Z . ثانيا ، المتغير الرابع فى جملة READ هو المتغير الحقيقى Z ولكن المواصفات المناظرة له فى جملة FORMAT هى الحقل الصحيح I 15

(ب) لاحظ أولا أن جملة FORMAT هواختصار للجملة

```
11 FORMAT(F8.1, F8.1, F8.1, I8, I8)
```

وبذلك فالمتغير الصحيح J أعطى مواصفات حقيقية F8.1 .

(جـ) أولا ، يجب ألا تكون هناك فصلة قبل A . ثانياً ، مجموع عرض الحقول هو 105 وهو كبير جداً على بطاقة البيانات التى بها 80 عمود فقط .

٣ — ٦ افترض أنه تم تثقيب بطاقة بيانات كما فى الشكل ٣ — ١٣ . أوجد قيم J و K و L كنتيجة لكل من الحالات الآتية :

(أ)
```
   READ(5, 11) J, K, L
11 FORMAT(I4, I3, I8)
```

(ب)
```
   READ(5, 12) J, K, L
12 FORMAT(I3, 2X, I6, 3X, I3)
```

(جـ)
```
   READ(5, 13) J, K, L
13 FORMAT(2I6, I2)
```

```
          1         2
12345678901234567890
1234567   33333
```

شكل ٣ — ١٣

لاحظ أولا أن الأعداد الصحيحة فقط هى التى تقرأ كمتغيرات حيث أن كل مواصفات الحقول الرقمية هى فى الصيغة Iw حيث w هى عرض الحقل .

(ا) عرض الحقول هو 4 و 3 و 8 على الترتيب . من ثم ، يخصص إلى J العدد الصحيح المثقب فى أول أربعة أعمدة (الأعمدة 1 إلى 4) ويخصص إلى K العدد الصحيح المثقب فى الأعمدة الثلاثة التالية (الأعمدة 5 إلى 7) ويخصص إلى L العدد الصحيح المثقب فى الأعمدة الثمانية التالية (الأعمدة 8 إلى 15) وبذلك تصبح :

L = 3333 و K = 567 و J = 1234

(ب) تأمر مواصفات الحقول الخمسة الحاسب بالتالى :

I3 خصص إلى J العدد الصحيح المثقب فى أول ثلاثة أعمدة (الأعمدة 1 إلى 3) : لذا تخصص 123 إلى J .

2X اترك العمودين التاليين (الأعمدة 4 إلى 5) ، حتى إذا كانت هناك معلومات مثقبة فى الأعمدة .

I6 خصص العدد الصحيح المثقب فى الأعمدة الستة التالية (أعمدة 6 إلى 11) إلى K حيث أن المسافات الخالية تفسر كأصفار فى الحقول الرقية . 670003 تخصص إلى K .

3X اترك الأعمدة الثلاثة التالية (الأعمدة 12 إلى 14) .

I3 خصص العدد الصحيح المثقب فى الأعمدة الثلاثة التالية (الأعمدة 15 إلى 17) إلى L وبذا تخصص 300 إلى L .

وبذلك تصبح : J = 123 و K = 670003 و L = 300

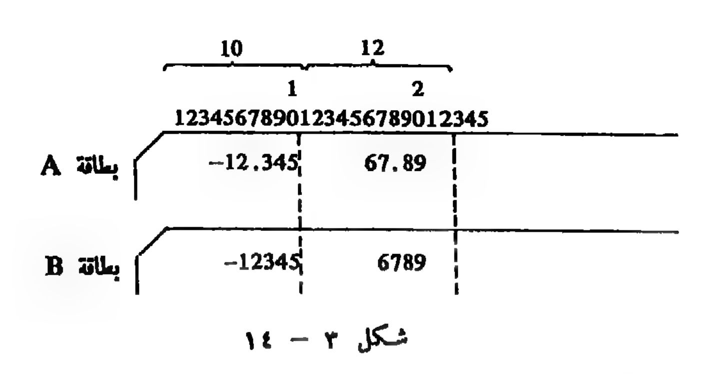
(جـ) 2I6 ، تكرر I6 مرتين ؛ أى مواصفات الحقول هى I6 ، I6 ، I2 بعرض الحقول 6 و 6 و 2 على الترتيب . وينتج عن هذا .

J = 123456 و K = 700033 و L = 33

٣ — ٧ افرض أن برنامجا به الجمل الآتية :

```
   READ(5, 21) A, B
21 FORMAT(F10.2, F12.3)
```

أوجد القيم المخصصة لكل من A و B إذا (أ) قرأت البطاقة A من الشكل ٣ — ١٤ ، (ب) قرأت البطاقة B من الشكل ٣ — ١٤ .



شكل ٣ — ١٤

لاحظ أولا أن عرض الحقول هو 10 و 12 على الترتيب ولذا يخصص الرقم الحقيقى الموجود فى الأعمدة 1 إلى 10 إلى A ويخصص الرقم الحقيقى الموجود فى الأعمدة 11 إلى 22 إلى B .

(أ) وحيث أن 12.345 — مثقبة فى الحقل الأول (الأعمدة 1 إلى 10) فهى تخصص إلى A وبما أن 67.89 مثقبة فى الحقل الثانى (الأعمدة 11 إلى 22) فهى تخصص إلى B . حيث أن الرقين لهما علامات عشرية فيمكن أن تهمل 2 فى F10.2 وتهمل 3 فى F12.3 .

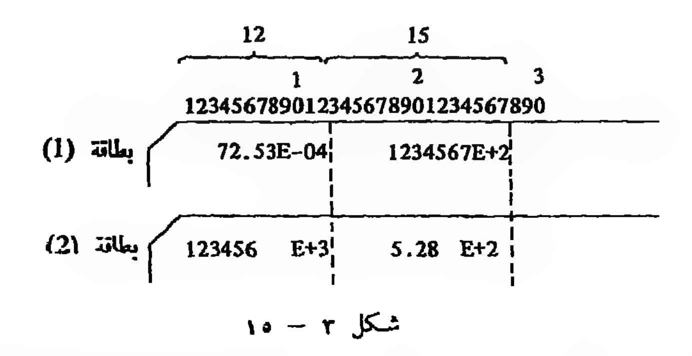
(ب) لاحظ عدم وجود علامة عشرية مثقبة فى الحقل الأول (الأعمدة 1 إلى 10) لذلك ستتسبب مواصفات الحقل F10.2 فى إضافة علامة عشرية بعد مكانين من الجانب الأيمن للحقل . أى ما بين العمود 8 والعمود 9 . ومن ثم تخصص

123.45 — إلى A . أيضاً العدد المثقب فى الحقل الثانى (الأعمدة 11 إلى 22) ليس به علامة عشرية ، لذا فالموصفات F12.3 سوف تضيف علامة عشرية بعد ثلاثة أماكن من الجانب الأيمن للحقل ، أى ما بين العمود 19 والعمود 20 . ومن ثم ، تخصص 678.9 إلى B . وأننا نؤكد أن موضع الرقم فى الحقل الخاص به سيختلف إذا ثقب بدون علامة عشرية

٣ — ٨ افرض أن برنامجاً به الجمل الآتية :

```
   READ(5, 22) A, B
22 FORMAT(E12.4, E15.6)
```

أوجد القيم المخصصة لـ A و B إذا كانت بطاقة البيانات (أ) بطاقة (1) من الشكل ٣ — ١٥ ، (ب) بطاقة (2) من الشكل ٣ — ١٥ .



شكل ٣ — ١٥

لاحظ أن عرض الحقول هو 12 و 15 على الترتيب ؛ من ثم ، يخصص الرقم الحقيقى الموجود فى الأعمدة 1 إلى 12 إلى A ويخصص الرقم الحقيقى الموجود فى الـ 15 عموداً التالية ، أى (الأعمدة 13 إلى 27) إلى B .

(أ) لاحظ أن 75.23E—04 مثقبه فى الحقل الأول (الأعمدة 1 إلى 12) ، ولذا تخصص إلى A . تهمل 4 فى مواصفات الحقل E12.4 حيث أن الرقم مثقب بعلامة عشرية . وفى الحقل الثانى (الأعمدة 13 إلى 27) ولأن الرقم مثقب بدون علامة عشرية لذا تدل 6 فى مواصفات الحقل E15.6 الحاسب بأن يضيف علامة عشرية بعد ستة أماكن من يسار الحرف E أى ما بين العمود 8 و 9 . وبذلك تخصص 1.23456 E + 2 إلى B .

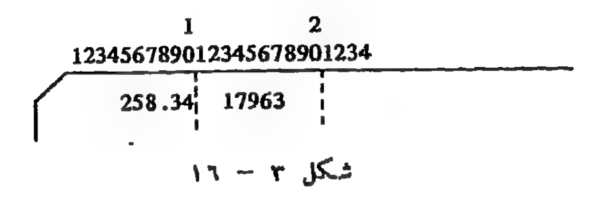
(ب) لاحظ أنه ليس هناك علامة عشرية مثقبة فى الحقل الأول ، بالتالى ستتسبب 4 فى مواصفات الحقل E12.4 فى إضافة علامة عشرية ما بين العمود الرابع والخامس إلى يسار الحرف E ، أى ما بين العمود 5 و 6 . من ثم تخصص 12345.6 E + 3 إلى A . وحيث أن الرقم المثقب فى الحقل الثانى به علامة عشرية لذلك تعامل المسافة الخالية فى العمود 27 كصفر . لذلك تخصص 5.28 E + 20 (وليس 5.28 E + 2) إلى B .

٣ — ٩ افرض أن بطاقة بيانات هى كما فى الشكل ٣ — ١٦ . أوجد قيم المتغيرات كنتيجة لكل من أزواج الجمل READ — FORMAT الآتية :

(أ)
```
   READ(5, 21) A, K
21 FORMAT(F10.1, I10)
```

(ب)
```
   READ(5, 22) A, B
22 FORMAT(F10.3, F10.6)
```

(جـ)
```
   READ(5, 23) A, B
23 FORMAT(E10.3, E10.4)
```



شكل ٣ - ١٦

(أ) بطاقة المواصفات تبين أن عرض الحقلين 10 و 10 على الترتيب وتأمر الحاسب بالآتى :

F10.1 تخصص الرقم الحقيق المثقب فى أول 10 أعمدة (الأعمدة 1 إلى 10) إلى A لذا تصبح A = 258.34 وتهمل 1 فى F10.1 حيث أن الرقم المثقب فى الحقل به علامة عشرية .

I 10 يخصص العدد الصحيح المثقب فى 10 أعمدة التالية (الأعمدة 11 إلى 20) إلى K حيث تقرأ المسافات الخالية كأصفار ، لذا تصبح K = 17963000

(ب) كما فى الجزء (أ) ، نحصل على A = 258.34 . تخبر مواصفات الحقل الثانى الحاسب أن يخصص الرقم الحقيق فى الأعمدة 11 إلى 20 إلى B حيث لا توجد علامة عشرية بداخل الحقل ، وتخبر 6 فى F10.6 الحاسب باضافة علامة عشرية على بعد ست مسافات من يسار العمود الأخير (عمود 20) . لذا نحصل على B = 17.963

(جـ) لاحظ أولا أن حرف E لا يظهر على بطاقة البيانات ، لذا تعامل مواصفات الحقول *Ew.d* نفس معاملة مواصفات الحقول *Fw.d* . تحتوى مواصفات الحقلين على عرض حقول 10 ، 10 على الترتيب وتأمر الحاسب بالآتى :

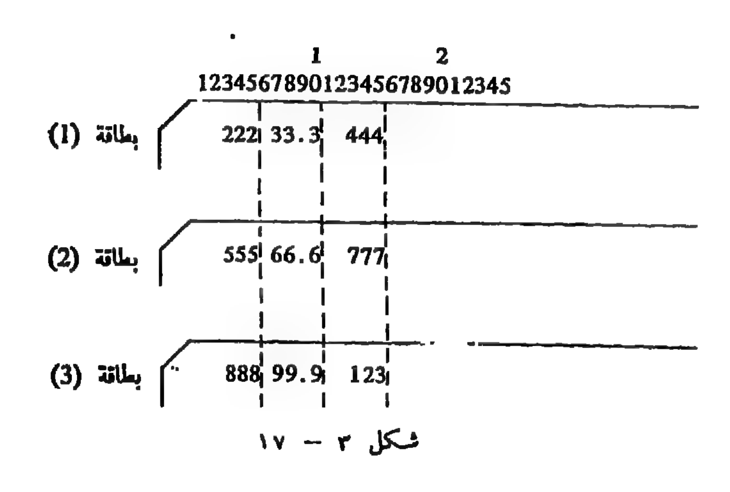
E10.3 تخصص الرقم الحقيق المثقب فى أول 10 أعمدة (الأعمدة 1 إلى 10) إلى A وحيث أن الحقل به علامة عشرية ، فتخصص 258.34 إلى A .

E10.4 تخصص الرقم الحقيق المثقب فى الأعمدة العشرة التالية (الأعمدة 11 إلى 20) إلى B حيث توجد علامة عشرية بداخل الحقل. تضاف علامة عشرية على بعد 4 مسافات من يسار العمود الأخير . لذا تصبح قيمة B = 1796.3

٣ - ١٠ افترض أنه تم تثقيب البطاقات الثلاث الأولى من مجموعة بطاقات البيانات كما فى الشكل (٣-١٧) . أوجد قيم المتغيرات لكل من الحالات الآتية :

(أ)
```
   READ(5, 11) J, A
   READ(5, 11) K, B, L, M
11 FORMAT(I5, F5.2, I5)
```

(ب)
```
   READ(5, 12) J, K, A, L, B, M
12 FORMAT(I5/I5, F5.2/I5, F5.2, I5)
```



شكل ٣ - ١٧

لاحظ أولا أن كل مواصفات الحقول بعرض 5 .

(أ) تذكر أن جملة READ أو WRITE لا يكتمل تنفيذها إلا بعد يتم تخصيص كل القيم / طبع كل المتغيرات المذكورة . أول جملة READ تقرأ من البطاقة الأولى المتغيرات I و J وتقرأ بمواصفات حقول I5 و F5.2 على الترتيب . ويكتمل تنفيذ جملة READ هذه ، وتترك البطاقة الأولى . تبدأ جملة READ الثانية أيضاً القراءة من البطاقة التى على القمة الآن ، أى البطاقة الثانية . حيث يوجد أربعة متغيرات وعددها أكثر من مواصفات الحقول (3) ، وبذلك تقرأ أكثر من سجل . فى هذه الحالة ، تقرأ بطاقتان . تقرأ المتغيرات K ، B ، L من البطاقة الثانية باستعمال I5 ، F5.2 ، I5 على الترتيب لتصبح K = 555 ، 666 ، L = 777 . ثم يقرأ المتغير M من البطاقة 3 باستعمال I5 ليصبح M = 888 وحيث قد تم تخصيص قيم لكل المتغيرات فقد اكتمل الآن تنفيذ جملة READ الثانية وتهمل البطاقة الثالثة .

(ب) حيث أن هناك شرطتين مائلتين ، فتقرأ على الأقل ثلاثة سجلات (بطاقات) . يقرأ المتغير J من البطاقة الأولى بـ I5 وهكذا تصبح J = 222 . تقرأ المتغيرات K و A من البطاقة الثانية بـ I5 ، F4.2 على الترتيب ، وهكذا K = 555 ، A = 66.6 وأخيراً L = 888 ، B = 99.9 ، M = 123 من البطاقة الثالثة .

المخرجات المصاغة

٣ – ١١ استخرج الأخطاء ، إن وجدت فى كل زوج WRITE–FORMAT :

(أ)
```
   WRITE(6, 31), A, B, N
31 FORMAT(F10.2, 3X, I8, 5X, I6)
```

(ب)
```
   WRITE(6, 32) J, Z, K,
32 FORMAT(1X, I10, 4X, 2F10.3)
```

(اعتبر عدم وجود جملة نوع قد غيرت نوع أى من المتغيرات)

(أ) لايجب أن تكون هناك فصلة قبل A ثم إعطاء المتغير الحقيقى B مواصفات الحقل الصحيح I8 .

(ب) لايجب أن تكون هناك فصلة بعد K . أيضاً تم إعطاء المتغير الصحيح K مواصفات الحقل الحقيقى F10.3 .

٣ – ١٢ إفرض أن K تحتوى 12345 وتم تنفيذ جملة WRITE التالية :

```
WRITE(6, 41) K
```

صف الخرج إذا كانت جملة FORMAT المصاحبة هى :

(أ) `41 FORMAT(I10)`
(ب) `41 FORMAT(4X, I8)`
(ج) `41 FORMAT(4X, I3)`
(د) `41 FORMAT(I5)`

(أ) حيث أن عرض الحقل 10 بذلك ستطبع قيمة مضبطة من الطرف الأيمن فى أول (9 = 1 – 10) تسع أعمدة من صفحة الطباعة ، كما فى الشكل ٣–١٨ (أ) . بمعنى آخر تتولد قيمة K مضبطة من الطرف الأيمن فى أول 10 حروف من سجل الإخراج ، أى تتولد bbbbb12345 لكن تستعمل أول مسافة ، للتحكم فى العربة ولا تطبع .

(ب) سيظهر الخرج كما فى الشكل ٣ – ١٨ (ب) حيث تأمر مواصفات الحقول الحاسب بالآتى :

4X تقدم سطراً واطبع (3 = 1 — 4) ثلاث مسافات فى أول ثلاثة أعمدة من صفحة الطباعة .

I8 اطبع قيمة K مضبطة من الطرف الأيمن فى الـ 8 أعمدة التالية (الاعمدة 4 إلى 11) بمعنى آخر ، تتولد أربع مسافات bbbb و12345bbb على الترتيب . فى سجل الإخراج ، ولكن يستعمل أول مسافة للتحكم فى العربة ولا تطبع .

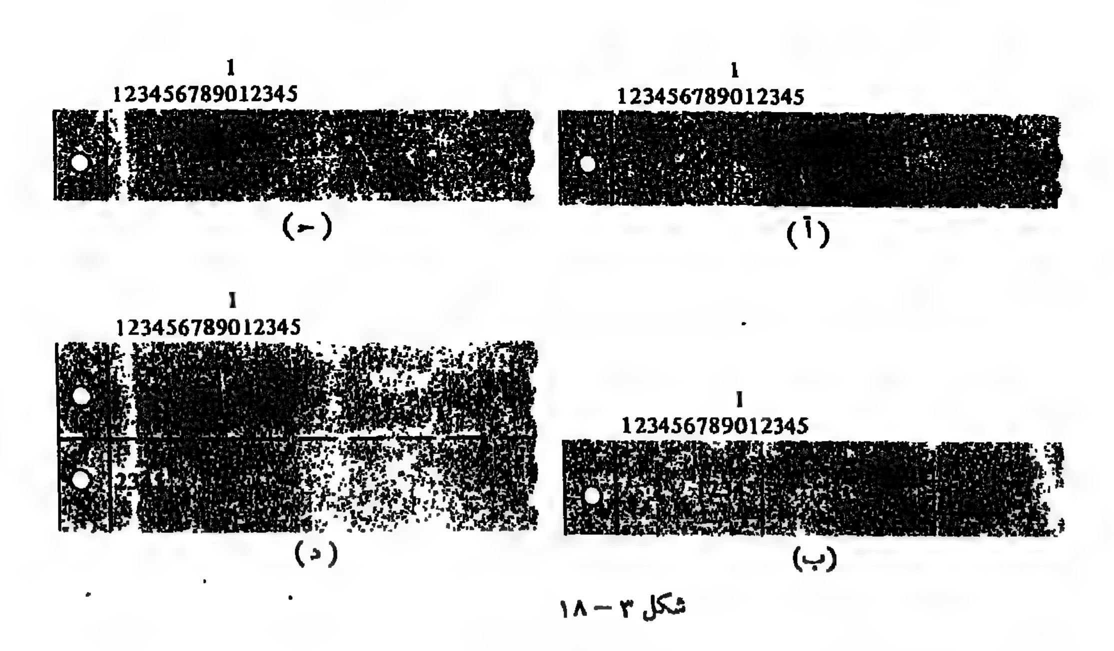
(جـ) تأمر مواصفات الحقول الحاسب بالآتى :

4X تقدم سطراً واحداً واطبع (3 = 1 — 4) ثلاث مسافات فى أول ثلاثة أعمدة من صفحة الطباعة .

I3 اطبع قيمة K مضبطة من الطرف الأيمن فى الأعمدة الثلاثة التالية (الأعمدة 4 إلى 6) .

حيث أن عرض الحقل المخصص لـ K أقل من عدد الخانات فى K ، إما أن تطبع ثلاث نجوم *** كما فى الشكل ٣ – ١٨ (جـ)، أو نبتر قيمة K من اليسار ، وبذلك تطبع 345 فقط .

(د) عرض الحقل I5 مساو لعدد الخانات فى K لذا تتولد 12345 فى سجل الإخراج . ولكن يستعمل أول حرف «1» التحكم فى العربة وتخبر وحدة الطباعة بأن تتقدم إلى قمة الصفحة التالية قبل طباعة الحروف الأربعة الباقية فى أول (4 = 1 — 5) أربعة أعمدة من صفحة الطباعة كما فى الشكل ٣ – ١٨ (د) .



شكل ٣ – ١٨

٣ – ١٣ افترض أن A تحتوى 135.2837 وتم تنفيذ جملة WRITE التالية :

```
WRITE(6, 42) A
```

صف الخرج إذا كانت جملة FORMAT المصاحبة هى :

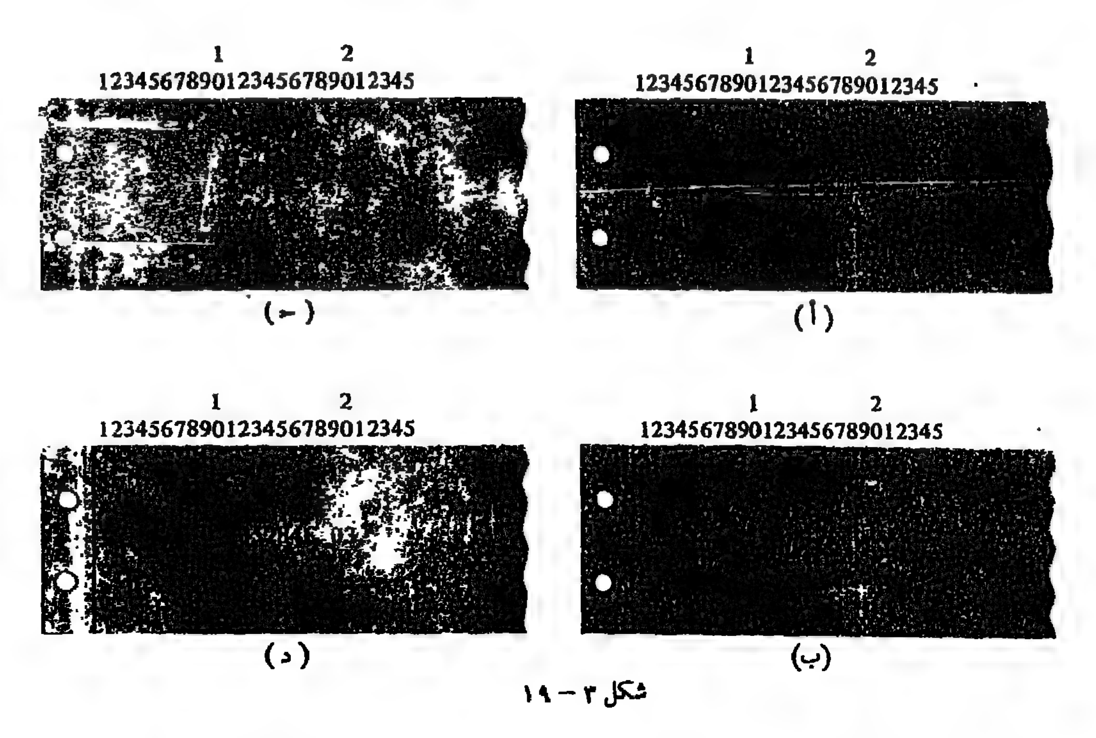
(أ) 42 FORMAT(6X, F10.1)

(ب) 42 FORMAT(6X, F12.2)

(جـ) 42 FORMAT(6X, F6.3)

(د) 42 FORMAT(6X, E15.6)

فى كل حالة ، تسبب مواصفات الحقل 6X تقدم وحدة الطباعة سطراً واحداً وأن تطبع مسافات فى أول (5 = 1 — 6) خمسة أعمدة (الأعمدة 1 إلى 5) من صفحة الطباعة . يظهر الخرج كما فى الشكل ٣ — ١٩ .



شكل ٣ — ١٩

وتأمر مواصفات الحقل الثانى الحاسب بالآتى :

(أ) تقريب قيمة A إلى مكان عشرى واحد ، أى ، إلى 135.3 وطباعة هذا الرقم مضبطاً من الطرف الأيمن فى الأعمدة العشرة التالية (الأعمدة 6 إلى 15) .

(ب) تقريب قيمة A إلى مكانين عشريين ، أى ، إلى 135.28 وطباعة هذا الرقم مضبطاً من الطرف الأيمن فى الإثنى عشر عموداً التالية (الأعمدة 6 إلى 17) .

(جـ) تقريب قيمة A إلى ثلاثة أماكن عشرية ، أى ، إلى 135.284 وطباعة هذا الرقم مضبطاً من الطرف الأيمن فى الأعمدة الستة التالية (الأعمدة 6 إلى 11) وحيث أن عرض الحقل(6) أقل من عدد الحروف (7) المراد طباعتها فإما تطبع ستة نجوم فى الحقل أو يبتر الرقم من اليسار بحيث تطبع 35.284 فى الحقل .

(د) تقريب قيمة A إلى ستة أرقام معنوية ، أى 135.284 وطباعة هذه القيمة فى الشكل الأسى القياسى ، أى 0.135284E03 مضبطاً من الطرف الأيمن فى الخمسة عشر عموداً التالية (الأعمدة 6 إلى 20) .

٣ — ١٤ افرض ID = 112233 و WAGE = 275.5 . حدد الخرج لكل زوج WRITE-FORMAT :

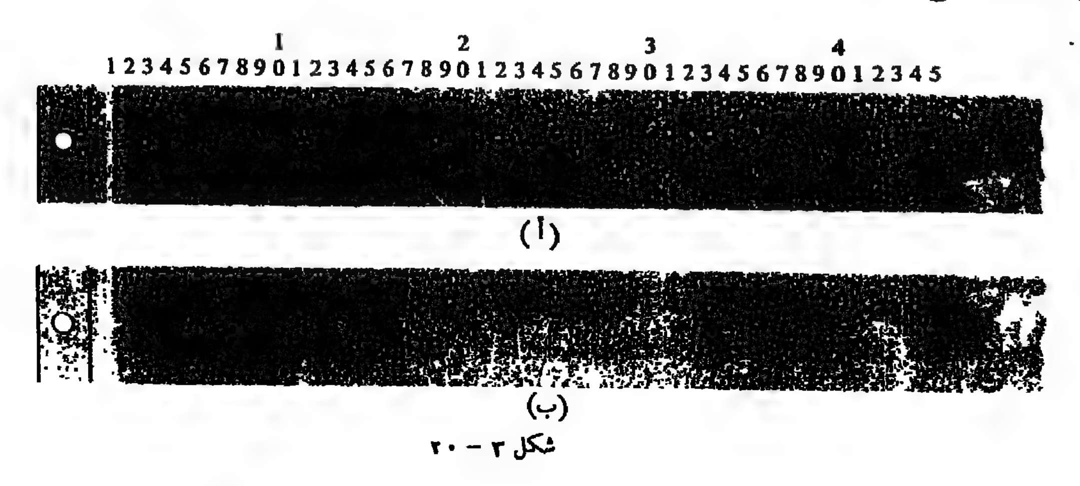
(أ)
```
   WRITE(6, 11) ID
11 FORMAT(6X, 'IDENTIFICATION NUMBER', 2X, I8)
   WRITE(6, 12) WAGE
12 FORMAT(6X, 'WEEKLY SALARY $', F8.2)
```
(ب)

(أ) تتقدم وحدة الطباعة سطراً واحداً وتطبع مسافات فى أول (5 = 1 — 6) خمسة أعمدة (الأعمدة 1 إلى 5) من صفحة الطباعة . يتسبب المدخل

'IDENTIFICATION NUMBER'

فى طباعة الرسالة بين الفصلات العليا فى الحقل التالى . حيث أن الرسالة تشغل 21 مسافة ، ستطبع فى عدد 21 عموداً التالية (الأعمدة 6 إلى 26) — تتسبب 2X فى طبع مسافات خالية فى الأعمدة 27 و28 ، وتتسبب I8 فى طباعة قيمة ID مضبطة من الطرف الأيمن فى الأعمدة الثمانية التالية (الاعمدة 29و36) . أنظر شكل ٣ — ٢٠ (أ) .

(ب) يظهر الخرج كما فى الشكل ٣ — ٢٠ (ب) .



(أ)

(ب)

شكل ٣ — ٢٠

٣ — ١٥ افرض D = 25367 ، M = 138 ، A = 256.174 ، B = 0.00285293 أوجد الخرج عند تنفيذ الزوج التالى WRIE-FORMAT :

(أ)
```
   WRITE(6, 11) ID, M, A, B
11 FORMAT(' ', I10, I6, F8.1, F10.3)
```

(ب)
```
   WRITE(6, 12) ID, A, B
12 FORMAT('1', I10, 2E15.5)
```

(أ) أول رمز فى سجل الإخراج هو مسافة معطاة بـ « » تمثل إشارة التحكم فى العربة ، ومن ثم يتقدم سطراً واحداً . مواصفات الحقول الأربعة الباقية عرضها 10 و 6 و 8 و 10 على الترتيب ، وتأمر الحاسب بالآتى :

I10 . وضع القيمة الصحيحة ID مضبطة من الطرف الأيمن فى أول عشرة أعمدة من صفحة الطباعة .

I6 طباعة قيمة M مضبطة من الطرف الأيمن فى الأعمدة الستة التالية (الأعمدة 11 إلى 16) .

F8.1 تقريب قيمة A إلى مكان عشرى واحد أى 256.2 ، وطباعة هذا الرقم الحقيقى مضبطاً من الطرف الأيمن فى الأعمدة الثمانية التالية (الأعمدة 17 إلى 24) .

F10.3 تقريب قيمة B أى 0.003 — إلى 3 أماكن عشرية ، وطباعة هذا الرقم الحقيقى مضبطاً من الطرف الأيمن فى الأعمدة العشرة التالية (الأعمدة 25 إلى 34) .

ومن ثم يظهر الخرج كما فى الشكل ٣ — ٢١ (أ) .

(ب) تأمر مواصفات الحقل الحاسب بالآتى :

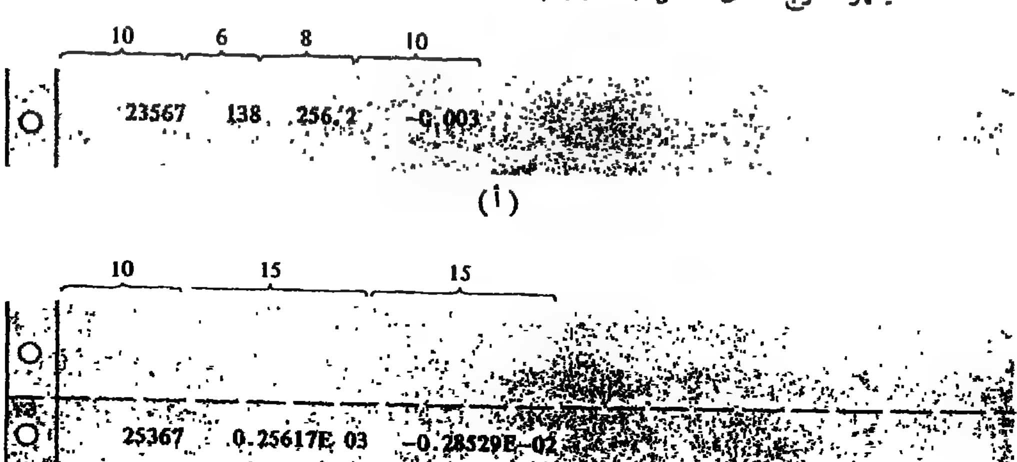
١٠ أمر التحكم و العربة التقدم إلى صفحة جديدة .

I10 طباعة قيمة ID فى أول عشرة أعمدة من صفحة الطباعة .

E15.5 تقريب قيمة المتغير A إلى 5 أرقام معنوية ، أى 256.17 وطباعة هذه القيمة فى الشكل الأسى القياسى . أى 0.25617E03 مضبطاً من الطرف الأيمن فى الخمسة عشر عموداً التالية (الأعمدة 11 إلى 25) .

E15.5 تقريب قيمة B إلى 5 أماكن معنوية ، أى 0.0028529– وطباعة هذا الرقم فى الشكل الأسى القياسى . أى 0.28529E–02– مضبطاً من الطرف الأيمن فى الخمسة عشر عموداً التالية (الأعمدة 26 – 40 .

يظهر الخرج كما فى الشكل ٣ – ٢١ (ب)



شكل ٣ – ٢١

٣ – ١٦ افرض أن المطلوب طباعة المتغيرات J ، A ، K و B بحيث يخصص لكل متغير صحيح 4 أعمدة ويخصص لكل رقم حـ 8 أعمدة بثلاثة أماكن عشرية وتفصل القيم عن بعضها بثلاث مسافات على الأقل . أوجد مدى القيم التى يمكن أن تطبع لكل متغير واكتب زوج WRITE-FORMAT الذى (أ) يطبع القيم على سطر واحد (ب) يطبع قيم J و A على سطر و K و B على سطر آخر .

يجب أن تقع قيم J و K بين 999– و 9999 ويجب أن تقع قيم A و B بين 999.999– و 9999.999

(أ)

```
   WRITE(6, 11) J, A, K, B
11 FORMAT(1X, I4, 3X, F8.3, 3X, I4, 3X, F8.3)
or
11 FORMAT(1X, 2(I4, 3X, F8.3, 3X))
```

(ب)

```
   WRITE(6, 11) J, A, K, B
11 FORMAT(1X, I4, 3X, F8.3)
```

٣ – ١٧ افرض أن J1 ، J2 ، J3 تحتوى على ثلاث درجات مواد دراسية ، وتحتوى K على متوسطاتها .

(أ) اكتب جزء البرنامج الذى سيطبع الآتى على صفحة جديدة مبتدأ من عمود 11 :

```
TEST SCORES
    XXX
    XXX
    XXX
  AVERAGE
    XXX
```

حيث تشير XXX'S إلى قسم درجات المواد الدراسية ومتوسطاتها .

(ب) نفذ ماسبق بجملة WRITE واحدة

(أ)

```
      WRITE(6, 11)
11    FORMAT('1', 10X, 'TEST SCORES')
      WRITE(6, 12) J1, J2, J3
12    FORMAT(1X, I17)
      WRITE(6, 13)
13    FORMAT(13X, 'AVERAGE')
      WRITE(6, 12) K
```

(ب)

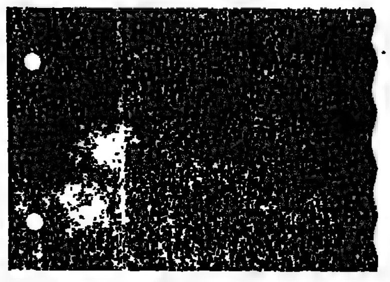
```
      WRITE(6, 21) J1, J2, J3, K
21    FORMAT('1', 10X, 'TEST SCORES'/1X, I17/1X, I17/1X, I17/13X, 'AVERAGE'/1X, I17)
or
21    FORMAT('1', 10X, 'TEST SCORES'/3(1X, I17/)13X, 'AVERAGE'/1X, I17)
```

٣ - ١٨ .فرض J1 = 111 ، J2 = 222 ، J3 = 333 ، J4 = 444 ، J5 = 555 حدد الخرج إذا تم تنفيذ الزوج الآتى :

```
      WRITE(6, 15) J1, J2, J3, J4, J5
15    FORMAT('0', I8/I9)
```

يتقدم الحاسب سطرين ، أى ، يتخطى سطر ('0') ويطبع J1 فى أول 8 أعمدة بالمواصفات (I8). تعلن الشرطة المائلة (/) عن نهاية السجل ، لذا تطبع J2 فى أول (8 = 1 — 9) ثمانى أعمدة من السطر التالى بمواصفات I9 . تتكرر جملة FORMAT لـ J3 و J4 وتتكرر مرة ثانية لـ J5 . تتسبب '0' فى تخطى سطر بين J2 و J3 وبين J4 و J5 . سيظهر الخرج كما فى الشكل ٣–٢٢

1
123456789012345



شكل ٣ – ٢٢

برامج بسيطة كاملة :

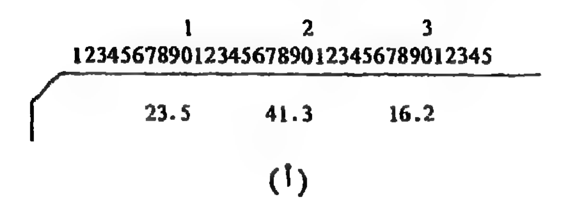
٣ - ١٩ تعطى المساحة السطحية (SUR) وحجم صندوق (VOL) ذا أبعاد a ، b ، c بالمعادلتين

$$SUR = 2(ab + ac + bc) \quad \text{و} \quad VOL = abc$$

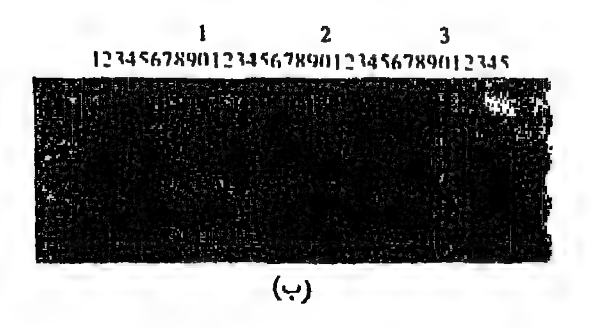
افترض أن أبعاد الصندوق هى 23.5 ، 41.3 ، 16.2 على الترتيب وتم تثقيبها على بطاقة بيانات كما فى الشكل ٣ – ٢٣

(أ) أوجد الخرج للبرنامج التالى :

```
C
C        SURFACE AREA AND VOLUME OF A BOX
C
         READ(5, 61) A, B, C
    61   FORMAT(3F10.2)
         SUR = 2.0(A*B + A*C + B*C)
         VOL = A*B*C
         WRITE(6, 62) A, B, C, SUR, VOL
    62   FORMAT(6X, 'DIMENSIONS OF BOX'/7X, 3(F5.1, 1X)//
     1           6X, 'SURFACE AREA =', F10.1//6X, 'VOLUME =', F16.2)
         STOP
         END
```



(أ)



(ب)

شكل ٣ – ١٣

تنفيـــذ الـــبرنامج

يقرأ الحاسب قـــيم A ، B ، C تبعاً لجملة FORMAT المرقة 61 والتى هى اختصار للآتى :

61 FORMAT(F10.2, F10.2, F10.2)

فى كل حالة ، عرض الحقل هو 10 لذا تخصص الأرقام المثقبة فى (الأعمدة 1 إلى 10 ، 11 إلى 20 ، 21 إلى 30) إلى A و B و C على الترتيب . وبهذا تصبح A = 23.5 ، B = 41.3 ، C = 16.2 (تهمل 2 فى 10.2 حيث أن الأرقام على بطاقة البيانات بها علامات عشرية فعلا) .

يحسب الحاسب 2.0*(A*B + A* C + B*C) وهى 4040.62 ويخصص هذا الرقم إلى SUR ويحسب A*B*C وهى 15722.910 ويخصص هذا الرقم إلى VOL :

SUR ← 4040.62, VOL ← 15722.910

يأمر الزوج WRITE-FORMAT الحاسب بطباعة الخرج كما فى الشكل ٣–٢٣ (ب) . لاحظ أن شرطة مائلة واحدة / تسبب تقدم وحدة الطباعة إلى السطر التالى ، ولكن شرطتان مائلتان تسببان تقدم وحدة الطباعة سطران ، ومن ثم ، تتخطى سطراً ، لاحظ أيضاً أن SUR تم تقريبها إلى مكان عشرى واحد (F10.1) ، وVOL تم تقريبهما إلى مكانين عشريين (F16.2) .

٣ – ٢٠ بافتراض أن $D = a_1b_2 - a_2b_1 \neq 0$ ، فان حل المعادلتين الخطيتين

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2$$

يعطى بواسطة

$$x = \frac{b_2c_1 - b_1c_2}{a_1b_2 - a_2b_1} \qquad y = \frac{a_1c_2 - a_2c_1}{a_1b_2 - a_2b_1}$$

اكتب برنامج فورتران يقرأ قيم A1 و B1 و C1 من بطاقة بيانات واحدة ثم قيم A2 و B2 و C2 من بطاقة بيانات ثانية :
أحسب القيمتين X ، Y ثم اطبع الخرج حتى يظهر كالآتى :

```
COEFFICIENTS OF LINEAR EQUATIONS
     FIRST      *****     *****     *****
     SECOND     *****     *****     *****
SOLUTION OF LINEAR EQUATIONS
     X = ********
     Y = ********
```

```
C
C       SOLUTION OF LINEAR EQUATIONS
C
        READ(5, 11) A1, B1, C1, A2, B2, C2
   11   FORMAT(3F10.1)
        D = A1*B2 - A2*B1
        X = (B2*C1 - B1*C2)/D
        Y = (A1*C2 - A2*C1)/D
        WRITE(6, 12) A1, B1, C1, A2, B2, C2, X, Y
   12   FORMAT(11X, 'COEFFICIENTS OF LINEAR EQUATIONS'//15X, 'FIRST',
     1          3X, 3F8.1/15X, 'SECOND', 2X, 3F8.1//
     1          11X, SOLUTION OF LINEAR EQUATIONS'//15X, 'X =',
     1          2X, F12.2/15X, 'Y =', 2X, F12.2)
        STOP
        END
```

مسائل تكميلية

إدخال / إخراج غير مصاغ

٣ — ٢١ اكتشف الأخطاء ، أن وجدت ، فى جمل I/O غير المصاغه

(ا) READ, FIRST, LAST, NEXT,

(ب) PRINT ID, WAGE, RATE,

(جـ) READ, INT, LOT, AREA

(د) PRINT, A, B, C, D,

٣ — ٢٢ افرض أن أول أربع بطاقات من مجموعة بطاقات البيانات تم تثقيبها كالآتى :

أول بطاقة : 11 ,22 ، 3.3 ، 4.4

ثانى بطاقة : 5.5 ، 6.6 ، 77

ثالث بطاقة : 88 ، 99 ، 2.34

رابع بطاقة : 5.67 ، 8.90 ، 123 ,456

أوجد القيم المخصصة للمتغيرات لو نفذنا الآتى :

(أ)
```
READ, J, K, A  B, C
READ, L, M, X, Y
```

(ب)
```
READ, J, K, A
READ, B, C, L, M
READ, X, Y
```

(جـ)
```
READ, J, K, A
READ, B, C, L
READ, M
READ, X, Y
```

٣ - ٢٣ افرض أن القيم 1.1 و 2.2 و 33 و 44 و 55 سوف يتم تخصيصها إلى المتغيرات A و B و J و K و L على الترتيب ، وباستعمال جملة READ التالية :

```
READ, A, B, J, K, L
```

بين كيف يجب أن تثقب البيانات على البطاقات إذا (أ) استعملت بطاقة واحدة فقط (ب) استعملت بطاقتان للبيانات مع وجود القيم الحقيقية على البطاقة الأولى والقيم الصحيحة على البطاقة الثانية .

٣ - ٢٤ إفرض أنه تم تثقيب بطاقتين للبيانات كالتالى :

أول بطاقة : 22.2 و 15

ثانى بطاقة : 25 و 35

أوجد المخرج إذا نفذ جزء البرنامج التالى :

(أ)
```
READ, A, J, K
B = A*6.0
PRINT, A, B, J, K
```

(ب)
```
READ, A
READ, J, K
B = A*6.0
PRINT, A, B, J
PRINT, K
```

(جـ)
```
READ, A, J, K
B = A*6.0
PRINT, A
PRINT, B, J, K
```

(د)
```
READ, A
READ, J, K
B = A*6.0
PRINT, A, J
PRINT, B
PRINT, K
```

الإدخال المصاغ

٣ - ٢٥ اكتشف الأخطاء ، إن وجدت فى كل زوج READ-FORMAT .

(أ)
```
   READ(5, 10), X, Y, L, M,
10 FORMAT(F8.1, F9.2, 2I12)
```

(ب)
```
   READ(5, 20) A, L, B, M
20 FORMAT(F10.3, I8)
```

(جـ)
```
   READ(5, 30) S, T, U, J, K
30 FORMAT(2F10.3, 3I8)
```

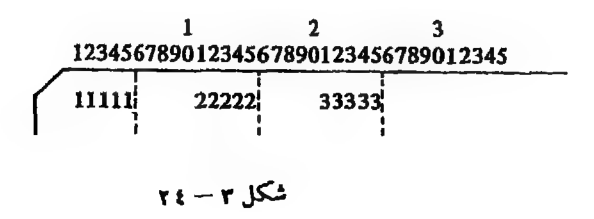
(د)
```
   READ(5, 40) A, J, B, K, C, L
40 FORMAT(F8.2, I6, F8.3)
```

٣ — ٢٦ افرض أنه تم تثقيب بيانات كما فى الشكل ٣ — ٢٤ ، وافرض أنه تم تنفيذ جملة READ الآتية :

```
READ(5, 50) J, K, L, M
```

أوجد القيم المخصصة لكل من J ، K ، L ، M إذا كانت جملة FORMAT المصاحبة هى :

(١) 50 FORMAT(I2, I3, I8, I3)
(ب) 50 FORMAT(4I4)
(c) (جـ) 50 FORMAT(2I3, 5X, 2I3)
(d) (د) 50 FORMAT(3I3, 2I4)

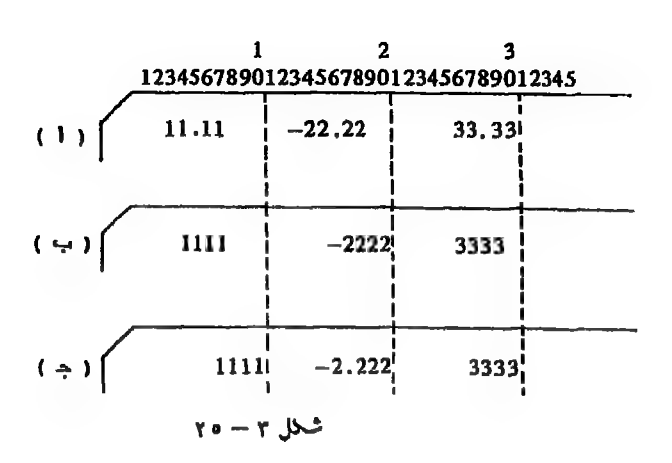


شكل ٣ — ٢٤

٣ — ٢٧ افرض أنه تم تنفيذ زوج READ-FORMAT الآتى :

```
      READ(5, 10) A, B, C
   10 FORMAT(F10.2, F10.3, F10.2)
```

أوجد القيم المخصصة لكل من A و B و C إذا كانت بطاقة البيانات كما هو موضح فى الشكل ٣ — ٢٥ .

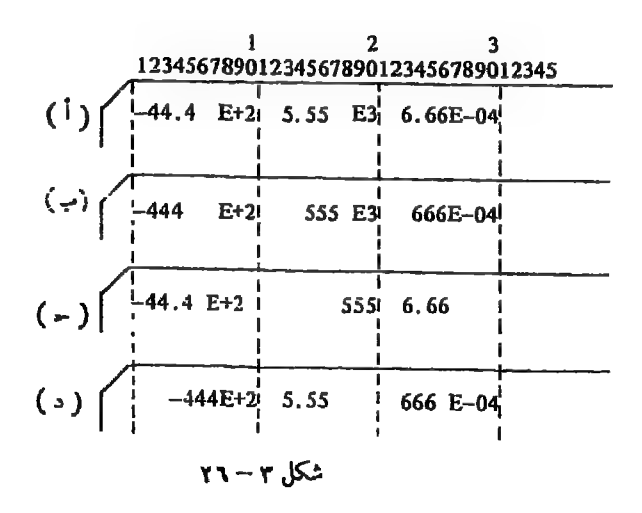


شكل ٣ — ٢٥

٣ — ٢٨ افرض أنه تم تنفيذ زوج READ-FORMAT الآتى :

```
      READ(5, 20) A, B, C
   10 FORMAT(E10.2, E10.3, E10.2)
```

أوجد القيم المخصصة لكل من A و B و C إذا كانت بطاقة البيانات كما هو موضح فى الشكل ٣ – ٢٦ .



شكل ٣ – ٢٦

٣ – ٢٩ افرض أنه تم تثقيب بطاقة بيانات كما فى الشكل ٣ – ٢٧ . أوجد القيم المخصصة للمتغيرات إذا تم تنفيذ زوج READ-FORMAT الآتى :

(أ)
```
   READ(5, 10) I, J, A, B
10 FORMAT(2I5, 2F5.1)
```
(ب)
```
   READ(5, 20) A, B, C, D
20 FORMAT(4F5.3)
```
(ﺟ)
```
   READ(5, 30) J, K, X, Y, Z
30 FORMAT(I4, 2X, I4, 3F4.2)
```
(د)
```
   READ(5, 40) R, S, T
40 FORMAT(3F4.1)
```
(ﻫ)
```
   READ(5, 50) J, A, B, L, M
50 FORMAT(I3, 2X, 2F4.1, I3, I4)
```
(و)
```
   READ(5, 60) I, J, K, X, Y, Z
60 FORMAT(2X, 3I1, 3X, F4.2, 2F3.1)
```

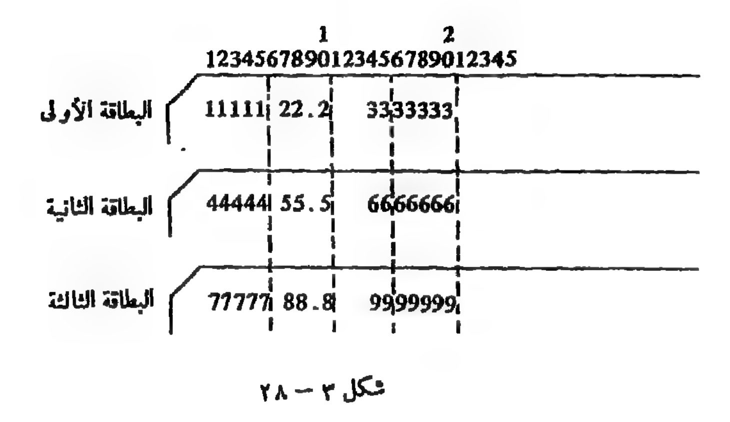
(ﺟ) (و)

```
          1         2
1234567890123456789012345
1234567890 1.11  222
```

شكل ٣ – ٢٧

٣ – ٣٠ افرض أنه تم تثقيب البطاقات الثلاث الأولى من مجموعة بيانات كما فى الشكل ٣ – ٢٨ . أوجد القيم المخصصة للمتغيرات إذا تم تنفيذ الآتى :

(أ)
```
   READ(5, 10) J, X
   READ(5, 10) K, Y, L, M
10 FORMAT(I5, F5.2, I5)
```
(ب)
```
   READ(5, 20) J, A, B, K
   READ(5, 20) L, C
20 FORMAT(I5, 2F5.2)
```
(ﺟ)
```
   READ(5, 30) J, A, B
   READ(5, 30) K, X, Y, L, M
30 FORMAT(I5, 2F5.2, I5)
```
(د)
```
   READ(5, 40) J, X
   READ(5, 40) K, Y, Z
40 FORMAT(I5, 2F5.2)
```
(ﻫ)
```
   READ(5, 50) J, X, Y, Z, K, L, M
50 FORMAT(I5, 2F5.2/F5.2/I5, 5X, 2I4)
```



شكل ٣ – ٢٨

الخرج المصاغ

٣ – ٣١ اكتشف الأخطاء ، أن وجدت ، فى كل زوج WRITE-FORMAT

(أ)
```
      WRITE(6, 10) X, Y, M, N,
   10 FORMAT(1X, 3F8.1, 2I9)
```

(ب)
```
      WRITE(6, 20) A, L, B, M
   20 FORMAT(1X, F9.2, 2(I8, E7.3)),
```

(جـ)
```
      WRITE(6, 30), S, T, U, J, K
   30 FORMAT(2E12.3, 2(F8.1, 3X, I7))
```

(د)
```
      WRITE(6, 40) A, J, B, C, K
   40 FORMAT(F7.1, I8, E6.2)
```

٣ – ٣٢ افرض أن J و K و L تحتوى 1111 و2222 و3333 على الترتيب وتم تنفيذ

```
      WRITE(6, 50) J, K, L
```

صف الخرج إذا كانت جملة FORMAT المصاحبة هى :

(أ) `50 FORMAT(3I10)`

(ب) `50 FORMAT(1X, I10/2I10)`

(جـ) `50 FORMAT(1X, 'THE NUMBERS ARE', 3I8)`

(د) `50 FORMAT(1X, I10)`

(هـ) `50 FORMAT(1X, 2I10)`

(و) `50 FORMAT(I4, 7X, I3, 7X, I4)`

٣ – ٣٣ افرض أن A و B و C تحتوى 111.222 و 444.666 و 777.888 على الترتيب ، وتم تنفيذ :

```
      WRITE(6, 60) A, B, C
```

صف الخرج إذا كانت جملة FORMAT المصاحبة هى :

(أ) `60 FORMAT(1X, 3F10.2)`

(ب) `60 FORMAT(1X, F10.1, 2X, F5.3, 2X, F10.1)`

(جـ) `60 FORMAT('1', E15.5, E15.2, E15.4)`

(د) `60 FORMAT(1X, 2E12.4)`

(هـ) `60 FORMAT(1X, F10.2///1X, E10.3///1X, F10.1)`

(و) `60 FORMAT(F7.3, 2X, E15.7/F15.1)`

٣ - ٣٤ افرض أنه تم تنفيذ :

(١) WRITE(6, 70) J, K, X, Y

مع فرض أن كل عدد صحيح سيخصص له خمسة أعمدة ، وأن كل رقم حقيق سيخصص له سبعة أعمدة مع مكانين عشريين . سوف يفصل بين المتغيرات ثلاث مسافات على الأقل . أوجد مدى القيم التى يمكن أن تطبع لكل متغير . وأكتب جملة FORMAT المصاحبة التى :

(أ) تطبع القيم على سطر واحد

(ب) تطبع القيم على أول سطر لصفحة جديدة .

(حـ) تطبع القيم على السطر الخامس لصفحة جديدة .

(د) تطبع الأعداد الصحيحة على سطر والأرقام الحقيقية على السطر التالى .

(هـ) تطبع كل رقم على سطر مختلف .

(و) تطبع الأعداد الصحيحة على ثالث سطر من صفحة جديدة والأرقام الحقيقية على السطر السادس من نفس الصفحة .

برامج :

٣ - ٣٥ افرض أن الضريبة الفيدرالية FTAX هى 15 فى المائة منضريبة الولاية STAX هى 2.5 فى المائة من إجمالى الدخل WAGE افرض بطاقة تحتوى على رقم ID للموظف (9خانات) ثم على أجره فى الساعة RATE وعدد الساعات HOUR التى اشتغلها خلال الأسبوع . اكتب برنامج يطبع الـ ID ، HOUR ، WAGE، FTAX، STAX والأجر الصافى للموظف TPAY.

٣ - ٣٦ افرض x و y و z هى الدوال الآتية فى t .

$$x = t^3 - 8t + 4, \qquad y = \sin t + \cos 2t, \qquad z = e^{3t+5}$$

اكتب برنامجاً يقرأ t ، ويطبع t و x و y و z

٣ - ٣٧ افرض أن سيارة تحركت من سكون بعجلة ثابتة a لعدد من الثوانى . وأن السرعة النهائية v والمسافة d التى سافرتها السيارة تعطى بالقوانين :

$$d = \tfrac{1}{2}at^2 \qquad \text{and} \qquad v = at$$

اكتب برنامجاً يقرأ a و t ويطبع t و d و v

٣ - ٣٨ اكتب برنامجاً يقرأ الارتفاع h وأطوال القاعدتين b_1 و b_2 لشبه منحرف T ويطبع h و b_1 و b_2 والمساحة T .

مسائل متنوعة :

٣ - ٣٩ افرض أن جملة READ الآتية تم تنفيذها :

READ(5, 10) A, B

أوجد عدد البطاقات المتخطاة قبل قراءة B إذا كانت جملة FORMAT هى :

(أ) `10 FORMAT(F10.2/////)`
(ب) `10 FORMAT(F10.2/////F10.2)`

[وبصورة عامة تتخطى عدد *n* من البطاقات إذا ظهر عدد *n* شرطات مائلة فى نهاية جملة FORMAT كما فى (أ) ، ولكن تتخطى عدد *n* — 1 من البطاقات إذا ظهر عدد *n* من الشرطات المائلة فى وسط جملة FORMAT كما فى (ب)] .

٣ — ٤٠ افرض أنه تم تنفيذ جملة WRITE الآتية :

```
WRITE(5, 20) A, B
```

أوجد عدد الأسطر الخالية قبل طباعة B إذا كانت جملة FORMAT هى :

(أ) `20 FORMAT(1X, F10.2/////)`
(ب) `20 FORMAT(1X, F10.2/////1X, F10.2)`

[وبصورة عامة تطبع عدد (*n* — 1) من الأسطر الخالية كلما ظهر عدد *n* من الشرطات المائلة فى نهاية أو فى وسط جملة FORMAT (قارن مع مسألة ٣ — ٢٩)]

اجابات للمسائل التكميلية المختارة

٣ — ٢١ (أ) لايجب أن تكون هناك فصلة بعد NEXT

(ب) يجب أن تكون هناك فصلة بعد PRINT ولا شيء بعد RATE.

(جـ) لايوجد أخطاء (صحيحة) .

(د) لايجب أن تكون هناك فصلة بعد D .

٣ — ٢٢

(أ) 11, 22, 3.3, 4.4, 5.5, 88, 99, 2.34, 5.67
(ب) 11, 22, 3.3, 5.5, 6.6, 77, 88, 5.67, 8.90
(جـ) 11, 22, 3.3, 5.5, 6.6, 77, 88, 5.67, 8.90

٣ — ٢٣ (أ) بطاقة البيانات : 1.1 و 2.2 و 33 و 44 و 55

(ب) أول بطاقة : 1.1 و 2.2

ثانى بطاقة : 33 و 44 و 55

٣ - ٢٤ (أ) 22.2 و133.2 و15 و25 على سطر واحد .

(ب) 22.2 و 133.2 و 15 على سطر و 25 على السطر التالى .

(جـ) 22.2 على سطر و 133.2 و 15 و 25 على السطر التالى .

(د) 22.2 و 25 على سطر 133.2 على سطر تال ، ثم 35 على السطر التالى .

٣ - ٢٥ (أ) لا يجب أن تكون هناك فصلة قبل X ولا بعد M .

(ب) لايوجد أخطاء (صحيحة) .

(جـ) يقابل المتغير الحقيقى U رمز الشكل الصحيح I8 .

(د) تتكرر جملة FORMAT لـ K و C و L ولكن رموز الشكل ليست من النوع المناسب (الصحيح) .

٣ - ٢٦

(أ) 111, 110, 222, 200 (جـ) 11, 111, 222, 220

(ب) 111, 110, 0, 222 (د) 1111, 1000, 22, 2220

٣ - ٢٧

(أ) 11.11, −22.22, 33.33 (ب) 11110.00, −2.222, 333.30 (جـ) 11.11, −2.222, 33.33

٣ - ٢٨ (أ) −44.4E+2, 5.55E+3, 6.66E−4 (جـ) −44.4E20, 0.555, 6.66

(ب) −4440.00E2, 5.550E3, 6.66E−4 (د) −4.44E2, 5.55, 66.60E−4

٣ - ٢٩ (أ) 12345, 67890, 1.11, 22.2 (جـ) 1234, 7890, 1.1, 10.02, 22.00 (هـ) 123, 678.9, 1.0, 110, 222

(ب) 12.345, 67.890, 1.11, 0.222 (د) 123.4, 567.8, 900.1 (و) 3, 4, 5, 90.01, 0.11, 0.2

٣ - ٣٠ (أ) 11111, 22.2, 44444, 55.5, 66, 77777 (د) 11111, 22.2, 44444, 55.5, 0.00

(ب) 11111, 22.2, 0.33, 44444, 77777, 88.8 (هـ) 11111, 22.2, 0.33, 444.44, 77777, 9, 9999

(جـ) 11111, 22.2, 0.33, 44444, 55.5, 0.66, 66666, 77777

٣ - ٣١ (أ) لايجب أن تكون هناك فصلة بعد N . يقابل المتغير الصحيح M رمز الشكل الحقيقى F8.1 .

(ب) لاتوجد أخطاء (صحيحة) .

(جـ) لايجب أن تكون هناك فصلة قبل S . يقابل المتغير الصحيح K رمز الشكل الحقيقى F8.1 .

(د) لاتوجد أخطاء (صحيحة) .

٣ - ٣٢ (أ) bbbbb1111bbbbbb2222bbbbbb3333

(ب) bbbbbb1111
bbbbb2222bbbbbb3333

(جـ) THE NUMBERS AREbbbb1111bbbb2222bbbb3333.

(د) تظهر الأرقام على ثلاثة أسطر مختلفة .

(هـ) لاتوجد إشارة صريحة للتحكم فى العربة لذا يستعمل أول حرف من « 1111 » للتحكم فى العربة ، وسيطبع فقط الثلاثة الباقية . أيضاً ، عرض حقل K ليس كافياً ، لذا أما ستطبع 222 أو *** وهكذا ، سيظهر واحد من الآتى على أول سطر من صفحة جديدة .

111bbbbbbb222bbbbbbb3333 أو 111bbbbbbb***bbbbbbb3333 (٣)

٣ - ٣٣ (أ) bbbb111.22bbbb444.67bbbb777.89

(ب) bbbbb111.2bb*****bbbbbbb777.9

(جـ) bbbb0.11122E 03bbbbbbb0.44E 03bbbbb0.7779E 03 على أول سطر من صفحة جديدة .

(د) bb0.1112E 03bb0.4447E 03, and then bb0.7779E 03 على السطر التالى .

(هـ) bbbb111.22 ثم b0·445E03 على السطر الرابع ثم bbbbb777.9 على السطر السابع .

(و) 11.222bbbb0.4446660E 03 على السطر الأول من صفحة جديدة . ثم bbbbbbbbb777.9 على السطر التالى .

٣ - ٣٤ يمكن أن تطبع J و K لقيم بين 9999 — و 99999 ويمكن أن تطبع X و Y لقيم بين 999.99 — و 9999.99

(ا) FORMAT(1X, 2(I5, 3X), 2(F7.2, 3X))
(ب) FORMAT('1', 2(I5, 3X), 2(F7.2, 3X))
(جـ) FORMAT('1'////1X, 2(I5, 3X), 2(F7.2, 3X))
(د) FORMAT(1X, I5, 3X, I5/1X, F7.2, 3X, F7.2)
(هـ) FORMAT(1X, I5/1X, I5/1X, F7.2/1X, F7.2)
(و) FORMAT('1'//1X, I5, 3X, I5///1X, F7.2, 3X, F7.2)

٣ - ٣٩ (أ) 5 (ب) 4

٣ - ٤٠ (أ) 4 (ب) 4

الفصل الرابع

نقل التحكم ، خرائط سير العمليات

٤ ـ ١ مقدمـــــة

عادة ينفذ الحاسب تعليمات برنامج الفورتران أمراً بعد الآخر إلا إذا أمر بغير ذلك . ويمكن التحكم فى ترتيب تنفيذ الأوامر بعدة تعليمات مختلفة . أساساً يوجد نوعان من نقل التحكم ، انتقال غير مشروط وانتقال مشروط . وسوف ندرس جمل التحكم هذه فى هذا الفصل ، والفصل الخامس ، والفصل الثانى عشر .

كلما تعقدت البرامج ، كلما أصبحت خريطة سير العمليات مساعداً كبيراً فى التخطيط والتصميم ، وبناء هيكل البرنامج . وخريطة سير العمليات هى تمثيل بيانى للخوارزم ، أى صورة مرئية لخطوات الخوارزم ، وأيضا لسير التحكم بين الخطوات المتعددة . (الخوارزم هى طريقة لحل المشكلة خطوة بخطوة) عموما ، فى خريطة سير العمليات نحيط كل عملية ، أو أمر أو سلسلة من الأوامر بصندوق ، ونشير إلى سير التحكم بأسهم موجهة بين الصناديق . وعلاوة على ذلك ،فإننا نشير إلى أنواع العمليات المختلفة بأشكال مختلفة من الصناديق كما هو مبين فى شكل ٤ ـ ١ . إذا استكملت خريطة سير العمليات فى صفحة أخرى ، أو إذا كان من الصعب وصل صندوقين ، نستعمل دائرة صغيرة مرقمة لتمثل هذا الاتصال .

بيضاوى للبداية أو النهاية

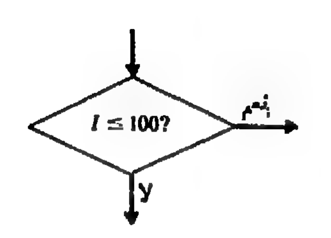
مستطيل لعملية حسابية أو أى عملية غير اتخاذ القرار .

معين لاتخاذ القرار

متوازى أضلاع للإدخال أو الإخراج

دائرة صغيرة للوصل

شكل ٤ ـ ١



يشير دائماً الصندوق ذو الشكل المعين إلى إتخاذ قرار ولذا سيخرج منه خطان أو أكثر. تعنون هذه الخطوط بنتائج القرار المختلفة أي بـ « نعم » أو « لا » أو بـ « حقيقي » أو « غير حقيقي » أو بـ « موجب أو « سالب » أو « صفر » . على سبيل المثال يمكن أن يظهر في خريطة سير العمليات السؤال (هل I ≤ 100 ؟) كما هو موضح على اليسار (وللتسهيل ، سنحذف عادة علامة الاستفهام)

عند هذه النقطة ، سنذكر مرة أخرى أن الرمز المقبول عالمياً لتخصيص قيمة لمتغير هو السهم ولكن ما يقابله في الفورتران هي العلامة = . وحيث أن خريطة سير العمليات مستقلة عن أي لغة ، فكثير من كتب الفورتران تفضل استعمال السهم في خرائط سير العمليات وسنتبنى هذه السياسة بصفة عامة . وعلى سبيل المثال ، للإشارة إلى أن N قد تم زيادتها بـ 1 أي خصصت القيمة N + 1 إلى N سنستعمل عادة

N ← N + 1

بدلا من

N = N + 1

أو أضف 1 إلى N

ونؤكد أن كل ما سبق مناقشته يستعمل بتوسع في كتب الفورتران .

٤ ــ ٢ الانتقال غير المشروط

يمكن إنجاز الانتقال غير المشروط بكتابة الجملة .

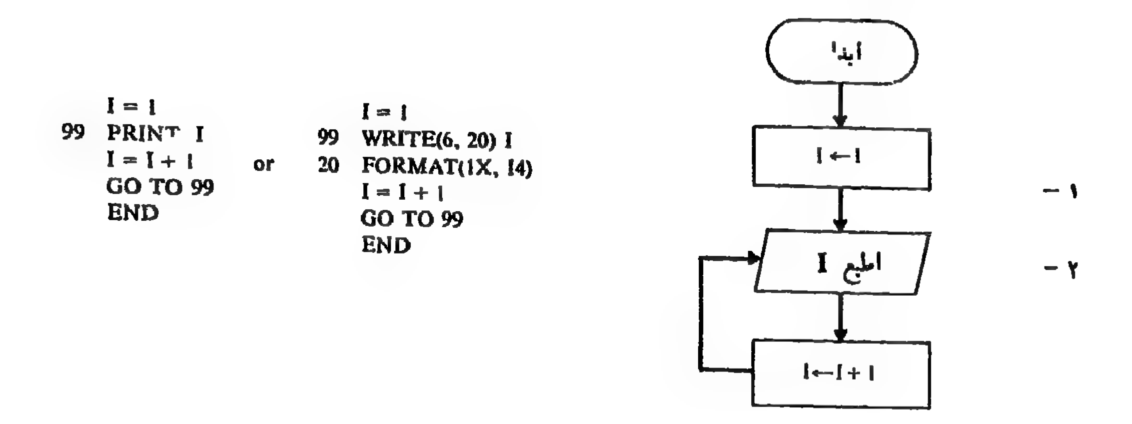
GO TO *n*

حيث *n* هي رقم جملة . وهذا يأمر الحاسب بأن يذهب ، بدون شرط ، إلى ذلك الجزء من البرنامج المبتدأ بالجملة التي تحمل الرقم *n* . واضح أن الجملة التي تحمل الرقم *n* يجب أن تكون جملة قابلة للتنفيذ . (انظر قسم ١ ــ ٨) .

مثال ٤ ــ ١

(ا) اكتب برنامجاً يولد ويطبع الأرقام الصحيحة الموجبة .....1, 2, 3 تلاحظ أن أي عدد صحيح موجب يمكن الحصول عليه بجمع 1 على الرقم السابق له . لذلك . فهي عملية تكرار ، تتم بجمع 1 على القيمة الحالية للعدد الصحيح . فإذا رمزنا إلى القيمة الحالية لعدد صحيح بالرمز I فإن العدد الصحيح التالي له (خليفه) هو I + 1 ، وحيث أن هذا العدد الصحيح سيستخدم مرة ثانية لحساب خليفة فسوف تخصص I + 1 إلى I

لبدء هذه العملية ، نعطي I قيمة ابتدائية وذلك بجعل I = 1 وفيما يلي خريطة سير العمليات لهذا البرنامج وما يقابلها في الفورتران :



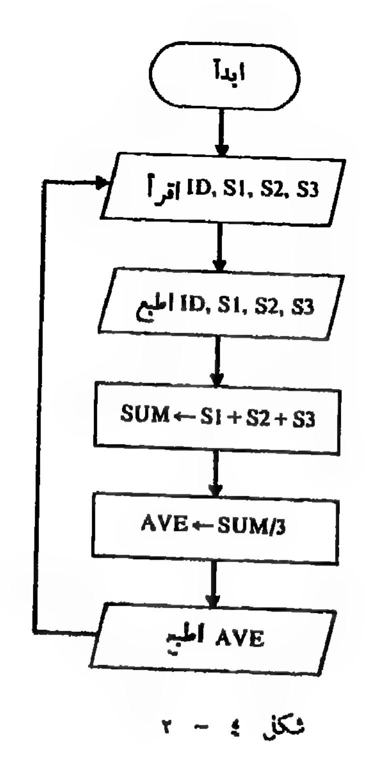

```
    I = 1
99  PRINT I
    I = I + 1
    GO TO 99
    END
```

or

```
    I = 1
99  WRITE(6, 20) I
20  FORMAT(1X, I4)
    I = I + 1
    GO TO 99
    END
```

(ب) تحتوى كل بطاقة من مجموعة البطاقات المعطاة على أربعة أرقام تمثل رقم تحقيق الشخصية(ID) التلميذ ، وكذا ثلاث درجات إختبار .
اكتب برنامجاً لحساب ، متوسط الاختبارات الثلاثة لكل بطاقة .

مرة ثانية ، نكرر عملية حساب المتوسط لثلاثة أرقام . لذلك نحتاج إلى انتقال غير مشروط . خريطة سير العمليات الخاصة بهذه العملية موضحة فى الشكل ٤ — ٢



شكل ٤ – ٢

وفيما يلي برنامج الفورتران ، المقابل لخريطة سير العمليات :

```
C
C        PROGRAM TO COMPUTE THE AVERAGE
C        FOR EACH STUDENT
C

   100   READ, ID, S1, S2, S3
         PRINT, ID, S1, S2, S3
         SUM = S1 + S2 + S3
         AVE = SUM/3.0
         PRINT, AVE
         GO TO 100
         END
```

وإذا استخدمنا مدخل/ خرج مصاغ يكون :

```
   100   READ(5, 10) ID, S1, S2, S3
    10   FORMAT(I5, F10.2, F10.2, F10.2)
         WRITE(6, 10) ID, S1, S2, S3
         SUM = S1 + S2 + S3
         AVE = SUM/3.0
         WRITE(6, 20) AVE
    20   FORMAT(1X, 'THE AVERAGE IS', F8.2)
         GO TO 100
         END
```

لاحظ أن فى كل من المثالين السابقين ، تتكرر العملية مراراً بدون توقف ويقال أن البرنامج كون حلقة تكرارية لا نهائية . ومن الواضح أننا نحتاج لبعض أوامر لاتخاذ القرارات لإنهاء العملية كما يجب . ذلك يقودنا إلى القسم التالى . ونشير هنا إلى أن قدرات الحاسب على اتخاذ القرار هى التى تجعل الحاسب فعال ، وفى منتهى القوة .

٤ — ٣ الانتقال المشروط

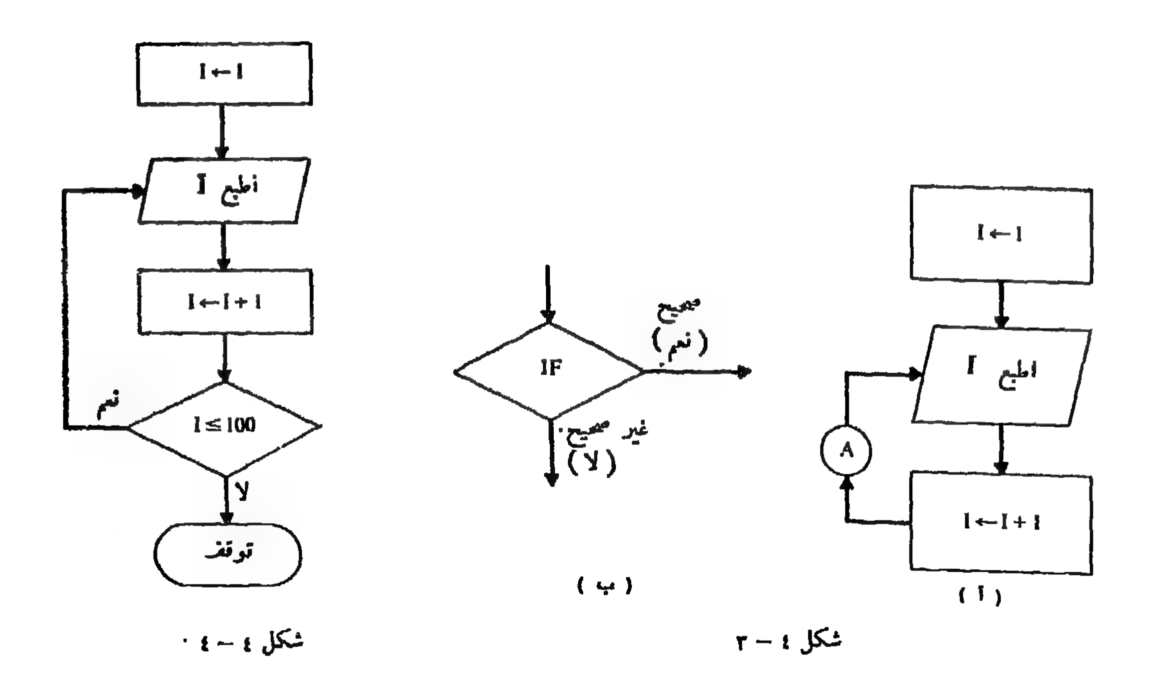
اعتبر مرة ثانية المثال ٤ — ١ (أ) لتوليد الأعداد الصحيحة . افرض أننا نريد أن نولد مائة عدد صحيح موجب فقط . أى ، افرض أننا نريد أن ننهى البرنامج بعد أن نولد العدد الصحيح 100 . إحدى الطرق للتوصل إلى ذلك هو أن نسأل السؤال التالى عند النقطة A من الشكل ٤ — ٣ (أ) . أى بعد تنفيذ I = I + 1 نسأل :

هل القيمة الحالية لـ I ≤ 100 ؟

إذا كانت الإجابة نعم ، كرر العملية ، وإلا فإنه العملية . بمعنى آخر يمكن أن يعطى الأمر التالى عند النقطة A وذلك باستخدام روابط اللغة الإنجليزية العادية :

IF <u>I ≤ 100</u> THEN <u>كرر العملية</u>
ELSE <u>توقف</u>

يمثل أمر ...IF....THEN...ELSE هذا بصندوق على شكل معين للقرار ، كما فى الشكل ٤ — ٣ (ب) خريطة سير العمليات التى تولد أول مائة عدد صحيح موجب معطاة فى الشكل ٤ — ٤ .



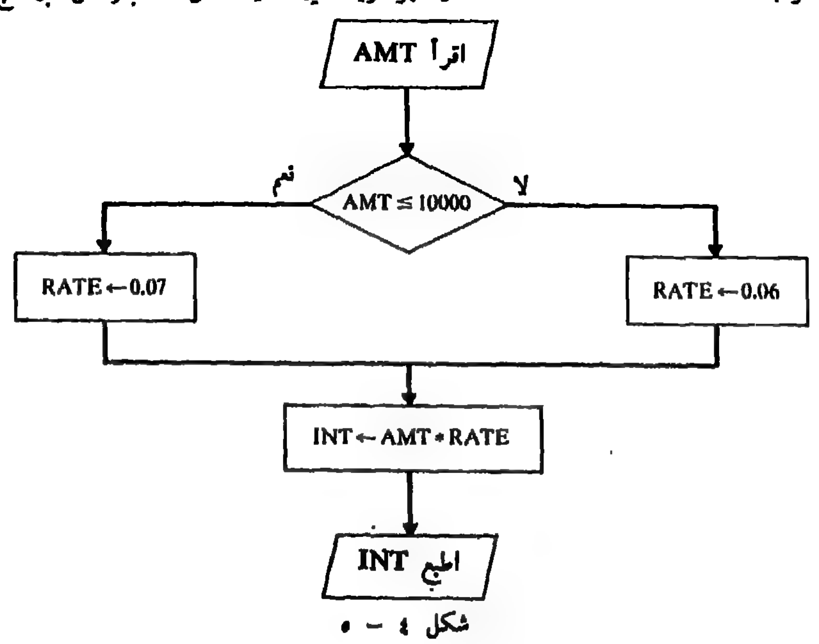
شكل ٤ – ٣

شكل ٤ – ٤ ·

اعتبر مثالا آخر ، افرض مقدار قرض AMT مثقبا على بطاقة ، وافرض أن RATE نسبة الفائدة هي 7 في المائة إذا كانت 10,000 $ ≥ AMT ، وتكون نسبة الفائدة 6 في المائة إذا كانت AMT أكبر من 10,000 $. ونريد أن نكتب جزء البرنامج لحساب الربح INT. نريد أن نعطي الأمر التالي بعد قراءة AMT :

```
IF   AMT ≤ 10,000   THEN   RATE = 0.07
                    ELSE   RATE = 0.06
```

مرة أخرى ، نرى الرابط ....IF...THEN..ELSE وتظهر خريطة سير العمليات لمثل هذا الجزء من البرنامج في الشكل ٤ – ٥ .



شكل ٤ – ٥

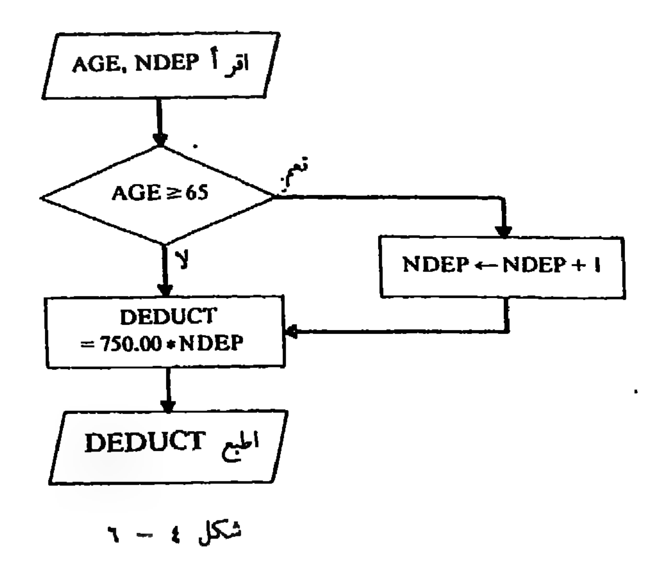
وعادة يفترض الرابط ....IF...THEN..ELSE وجود بديلين مختلفين كما هو موضح فى المثالين السابقين . وعلى أى حال فتوجد حالات تتطلب استعمال الرابط الإنجليزى IF...THEN ونبين ذلك بمثال ثالث

افرض أن عمر رجل AGE وعدد الذين يعولهم NDEP بيانات مثقبة على بطاقة . ونريد أن نحسب الضرائب المخصومة على دخله DEDUCT وهو 750 $ لكل معول وإذا كان الرجل عنده 65 سنة أو أكثر ، فيمكن أن يطالب بخصم إضافى (نفرض أنه لا يوجد أحد من معوليه عنده ٦٥ سنة أو أكثر) وبذلك فريد أن نعطى الحاسب الأمر التالى :

IF AGE ≥ 65 THEN NDEP = NDEP + 1

قبل حساب DEDUCT تظهر خريطة سير العمليات لمثل جزء البرنامج هذا فى الشكل ٤ — ٦

وفيما يلى معنى الجملة "IF X THEN Y" حينما تتحقق X نفذ Y أولا قبل أن تنتقل إلى الجملة التالية ، وحينما لا تتحقق X انتقل إلى الجملة التالية فوراً (أى تخطى Y) .



شكل ٤ — ٦

والفرق الدقيق بين الرابطين IF..THEN..ELSE و IF...THEN هو فرق غير مهم . ويناقش الفورتران الهيكل الذى يمكنه تنفيذ هذه الروابط فى قسم ٤ — ٥ وفى الفصل الثانى عشر .

٤ — ٤ تعبيرات مترابطة

وقبل أن نقدم كيف ينفذ الفورتران الانتقال المشروط ، سنبين أولا كيف نكتب بعض الجمل المترابطة بالفورتران .

يعطى جدول ٤ — ١ قائمة بستة معاملات ترابطية فى الفورتران تمثل العلاقات الرياضية الأكثر شيوعاً بين التعبيرات الحسابية . لاحظ أن كل مكافئ بالفورتران يتكون من أربعة حروف حرفان أبجديان تسبقهما وتتبعهما نقطة . وجود النقط ضرورى ، وإلا سيفسر الحاسب الحروف الأبجدية ، ولتكن LT على أنها متغير فضلا عن كونها معامل ترابط .

جدول ٤ — ١

فورتران	العلاقة
أقل من	.LT.
أقل من أن يساوى	.LE.
يساوى	.EQ.
لا يساوى	.NE.
أكبر من	.GT.
أكبر من أو يساوى	.GE.

يتكون التعبير المترابط من تعبيرين حسابين يصل بينهما معامل ترابط واحد . والتعبيرات المترابطة هى أبسط أشكال التعبيرات المنطقية (سنناقش تعبيرات منطقية تحتوى على أدوات منطقية أكثر تعقيداً فى الفصل التاسع) . وعند وقت التنفيذ يمثل كل تعبير منطقى ، وبالتالى كل تعبير مترابط شرطاً إما أن يتحقق أو لا يتحقق .

مثال ٤ — ٢

وفيما يلى تعبيرات رياضية والتعبيرات المترابطة المكافئة لها فى الفورتران :

$I < J$	I.LT.J
$(A + 3) > B$	(A + 3.0).GT.B
$5I = 2J$	5*I.EQ.2*J
$B \geq A^2$	B.GE.A**2
$(I + 3) \neq J$	(I + 3).NE.J
$(A + B^3) \leq 50$	(A + B**3).LE.50.0

افرض أن I و J و A و B تحتوى 2 و 5 و 3.0 و 4.0 على الترتيب . ومن ثم فالتعبيرات الثلاثة المترابطة الأولى تتحقق والثلاثة الأخرى لا تتحقق .

يجب أن نذكر هذين التعليقين :

١ — تجنب مقارنة تعبير صحيح بتعبير حقيقى لأن القيم الصحيحة تختلف عن القيم الحقيقية فى التمثيل الداخلى للذاكرة .

٢ — يجب أن نتوخى الحرص عند استعمال .EQ. و .NE. بين الأرقام الحقيقية . بسبب الخطأ النظرى فى الحسابات ، قد لا تتساوى إطلاقاً قيمتان حقيقيتان داخل الحاسب مع أن المفروض نظرياً أنهما متساويان . ولذا فبدلا من استخدام A.EQ.B فغالبا نختبر اختلافهما بشىء من التفاوت مثل

ABS(A — B).LT.0.0001

التعويض عن أخطاء التقريب الممكنة .

٤ — ٥ جملة IF المنطقية

يمكن أن تتم الانتقالات المشروطة بعدة طرق فى الفورتران وإحدى هذه الطرق هى استعمال جملة IF المنطقية وشكلها العام .

IF(*logexp*) Statement A

وبالتحديد ، تبدأ جملة IF المنطقية بكلمة IF ويتبعها تعبير منطقى *log exp* بين قوسين (تذكر أن التعبير المنطقى *log exp* هو تعبير له القيمة FALSE أو TRUE انظر أيضاً الفصل التاسع) وبعد ذلك تتبع الاقواس جملة قابلة للتنفيذ (Statement A) ولذلك فالجملة A يجب ألا تكون جملة IF أخرى أو جملة DO (سنناقش ذلك فى الفصل الخامس) .

فيما يلى أمثلة لجمل IF المنطقية :

```
IF(A.LE.B) GO TO 50
IF(L.GE.75) N = N + 1
IF(J.NE.K) WRITE(6, 30) X, Y.
```

وبطريقة رسمية ، فجملة IF المنطقية

```
IF(logexp) Statement A
```

لها المعنى التالى : إذا تحقق التعبير المنطقى (*Log exp*) أى (TRUE) فنفذ الجملة A وإلا تخطى الجملة A

وبأكثر دقة ، نفترض جزء من برنامج فورتران يحتوى

```
IF(logexp) Statement A
Statement B
```

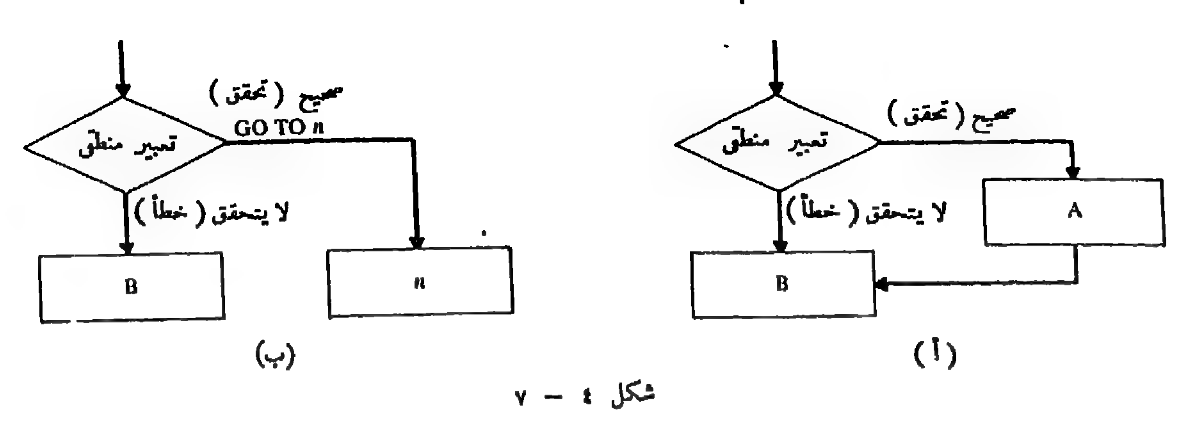
حيث جملة B هى أول جملة قابلة للتنفيذ تتبع جملة IF . وبذلك يمكن أن يحدث الموقفان التاليان :

١ – جملة A ليست جملة GO TO

٢ – جملة A هى جملة GO TO

الحالة ١

إذا كانت جملة A ليست جملة GO TO فجزء برنامج الفورتران السابق يعنى الآتى . عندما يتحقق التعبير المنطقى أى أن (Log exp) تكون TRUE نفذ الجملة A أولا ثم أكمل إلى الجملة التالية ، أى ، جملة B ؛ وعندما لا يتحقق التعبير المنطقى أى أن (Log exp) تكون FALSE ، أكمل فقط إلى الجملة التالية أى جملة B (متخطياً الجملة A) ونوضح هذا الموقف فى الشكل ٤ – ٧ (أ) . لاحظ أن جملة B تكون دائماً جملة منفذة . وتكافئ هذه الخريطة الاستعمال العادى للرابط الإنجليزى IF...THEN... الذى تمت مناقشته فى قسم ٤ – ٣ .



شكل ٤ – ٧

الحالة ٢

افرض أن جملة A هى جملة GO TO أى أن جزء برنامج الفورتران السابق هو

```
IF(logexp) GO TO n
Statement B
```

وهذا يعنى الآتى : إذا تحقق التعبير المنطقى أى أن (*Log exp*) تكون TRUE ينفذ جزء البرنامج المبتدأ بالجملة رقم n . أما إذا لم يتحقق التعبير المنطقى وأى أن (*Log exp*) تكون FALSE ينفذ جزء البرنامج مبتدأ بالجملة B . تظهر خريطة سير العمليات فى شكل ٤ – ٧ (ب) . وهى توضح أن بديلين مختلفين يمكن حدوثهما بعد القرار ، وهذا يكافئ الاستخدام العادى للرابط ...IF THEN...ELSE... وبذلك يمكن أن يستعمل هذا الشكل من جملة IF المنطقية كلما ظهر أ ..IF...THEN...ELSE فى الخوارزم (النظام الحسابى) . (انظر أيضاً الفصل الثانى عشر) .

والآن ادرس خريطة سير العمليات فى شكل ٤ – ٥ التى تحسب الربح INT على قرض AMT وتذكر أن خريطة سير العمليات تم رسمها كنتيجة للأمر .

```
IF   AMT ≤ 10,000   THEN   RATE = 0.07
                    ELSE   RATE = 0.06
```

نفرض أن جزء برنامج الفورتران

```
IF(AMT.LE.10000) RATE = 0.07
RATE = 0.06
```

قد استخدم لتنفيذ الأمر السابق . فتبعاً لشكل ٤ – ٧ (أ) والمناقشة التى تمت على الصفحة السابقة فإن RATE = 0.06 سوف تنفذ دائماً . ولكن ليس هذا هو ما نريد لذا لا يمكن استخدام هذا الجزء من برنامج الفورتران .

لكى نتخطى RATE = 0.06 حين يكون AMT ≤ 10,000 يجب أن نستخدم جملة GO TO فى جملة IF المنطقية كما فى برنامج الفورتران التالى :

```
      REAL INT
      READ(5, 10) AMT
   10 FORMAT(F15.2)
      IF(AMT.LE.10000.) GO TO 100
      RATE = 0.06
      GO TO 200
  100 RATE = 0.07
  200 INT = AMT*RATE
      WRITE(6, 20) INT
   20 FORMAT(1X, 'THE INTEREST IS', 2X, F10.2)
      STOP
      END
```

مثال ٤ — ٣

(أ) نترجم خريطة سير العمليات فى شكل ٤ — ٤ إلى الفورتران وهى تطبع أول 100 عدد صحيح موجب . لاحظ أن هناك طريقين متفرقين بعد القرار ولذلك فإننا نتوقع أن نستخدم الشكل IF *Log exp* GO TO ، من جملة IF المنطقية وفيما يلى هذه الترجمة .

```
    I = 1
99  WRITE(6, 10) I
10  FORMAT(1X, I5)
    I = I + 1
    IF(I.LE.100) GO TO 99
    STOP
    END
```

الجملتان

```
IF(I.LE.100) GO TO 99
STOP
```

هما ترجمة الأمر التالى إلى فورتران :

```
IF   I ≤ 100   THEN   GO TO 99 to repeat the process
                ELSE   STOP
```

(ب) وفيما يلى ترجمة خريطة سير العمليات ، التى تحسب الضرائب المخصومة على دخل شخص ، DEDUCT إلى فورتران :

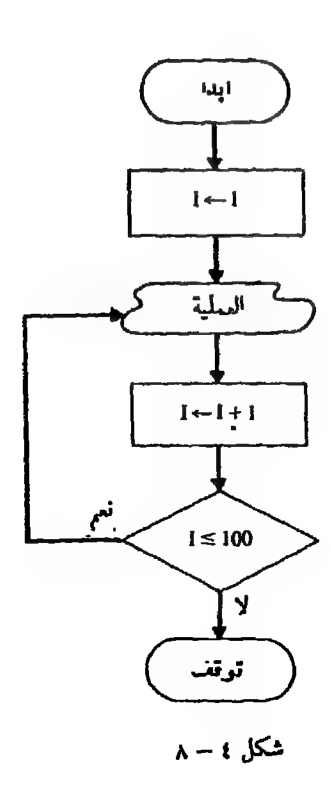
```
    INTEGER AGE
    READ(5, 10) AGE, NDEP
10  FORMAT(2I15)
    IF(AGE.GE.65) NDEP = NDEP + 1
    DEDUCT = 750.00*FLOAT(N)
    WRITE(6, 20) DEDUCT
20  FORMAT(1X, F12.2)
    STOP
    END
```

لاحظ التماثل بين الشكل ٤ — ٧ (ب) وجزء من الشكل ٤ — ٦ . لاحظ أيضاً أننا استعملنا الدالة المكتبية FLOAT لتحويل العدد الصحيح N إلى عدد حقيقى حتى نتجنب حساب النمط المختلط .

٤ — ٦ التحكم فى الحلقة التكرارية

افترض أننا نريد تكرار عملية (أى مجموعة من الأوامر) وليكن 100 مرة . يمكننا عمل ذلك باستخدام عداد I يعد عدد مرات تكرار العملية . أى أننا تخصص 1 العداد I فى البداية وبعد ذلك ، فى كل مرة ننفذ فيها العملية ، نزيد قيمة العداد I بمقدار

واحد (I) وننهى تنفيذ العملية إذا تعدت قيمة العداد I الرقم 100 ويوضح شكل ٤ — ٨ خريطة سير العمليات التى تنفذ هذه المهمة .



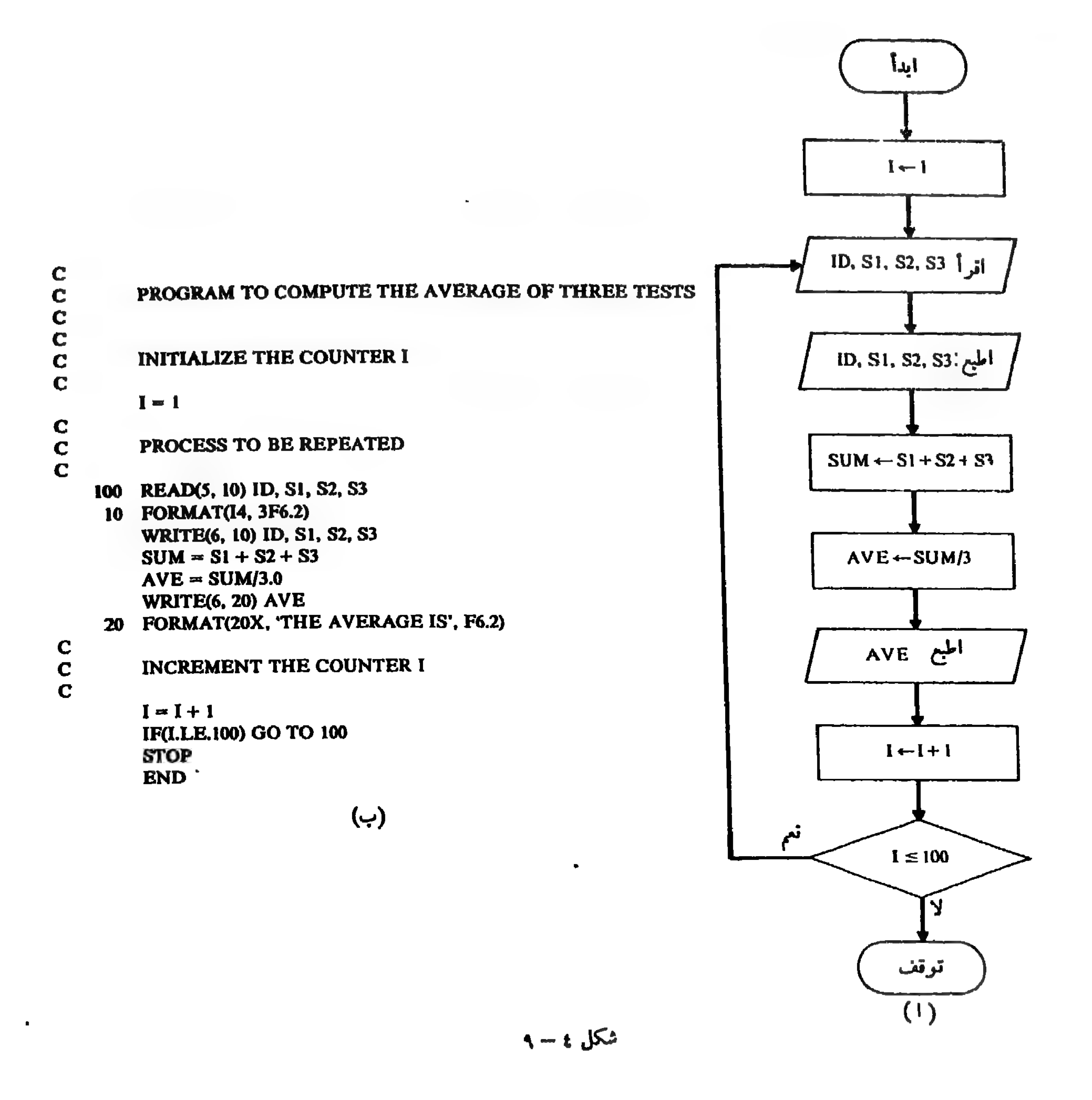
شكل ٤ — ٨

نلاحظ فى التو التماثل بين شكل ٤ — ٤ وشكل ٤ — ٨ . وبمعنى آخر فيمكن أيضاً أن نستخدم ميكانيكية التحكم التى استخدمناها لتوليد أول 100 عدد صحيح موجب كعملية العد والتحكم فى الخروج من حلقة تكرارية وذلك بتكرارها 100 مرة

مثال ٤ — ٤

استرجع مثال ٤ — ١ (ب) حيث أوجدنا متوسط ثلاث درجات لكل طالب . اكتب البرنامج إذا كان هناك 100 بطاقة فى مجموعة بطاقات البيانات .

نحن ندرج ببساطة خريطة سير العمليات لمثال ٤ — ١ (ب) فى المكان المناسب من شكل ٤ — ٨ لنحصل على شكل ٤ — ٩ (أ) . ترجمة الفورتران لبرنامجنا موضحة فى شكل ٤ — ٩ (ب) لاحظ أن ترجمة خريطة سير العمليات إلى فورتران عملية مباشرة عموما .




```
C
C        PROGRAM TO COMPUTE THE AVERAGE OF THREE TESTS
C
C
C        INITIALIZE THE COUNTER I
C
         I = 1
C
C        PROCESS TO BE REPEATED
C
  100    READ(5, 10) ID, S1, S2, S3
   10    FORMAT(I4, 3F6.2)
         WRITE(6, 10) ID, S1, S2, S3
         SUM = S1 + S2 + S3
         AVE = SUM/3.0
         WRITE(6, 20) AVE
   20    FORMAT(20X, 'THE AVERAGE IS', F6.2)
C
C        INCREMENT THE COUNTER I
C
         I = I + 1
         IF(I.LE.100) GO TO 100
         STOP
         END
```

(ب)

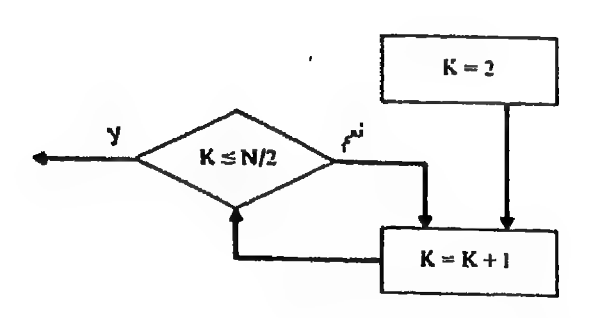
شكل ٤ — ٩

مثال ٤ — ٥

افرض أننا نريد كتابة برنامج لقراءة عدد صحيح $N > 2$ وتحدد ما إذا كانت N عدد أولى أم لا .

تذكر أن N تكون عددا أولياً إذا كانت لا تقبل القسمة إلا على نفسها وعلى الواحد الصحيح على سبيل المثال والاعداد 2 و 3 و 5 و 13 و 29 و 67 أعداد أولية . ولكن 35 ليس عدداً أولياً حيث أنه يقبل القسمة على 5 ، 7 . ومن الواضح أنه إذا كانت N ليست عدد أولياً فإن N تقبل القسمة على أحد هذه الأعداد الصحيحة 2 و 3 و ... N/2 (قارن المسألة ٤ – ٤٨)

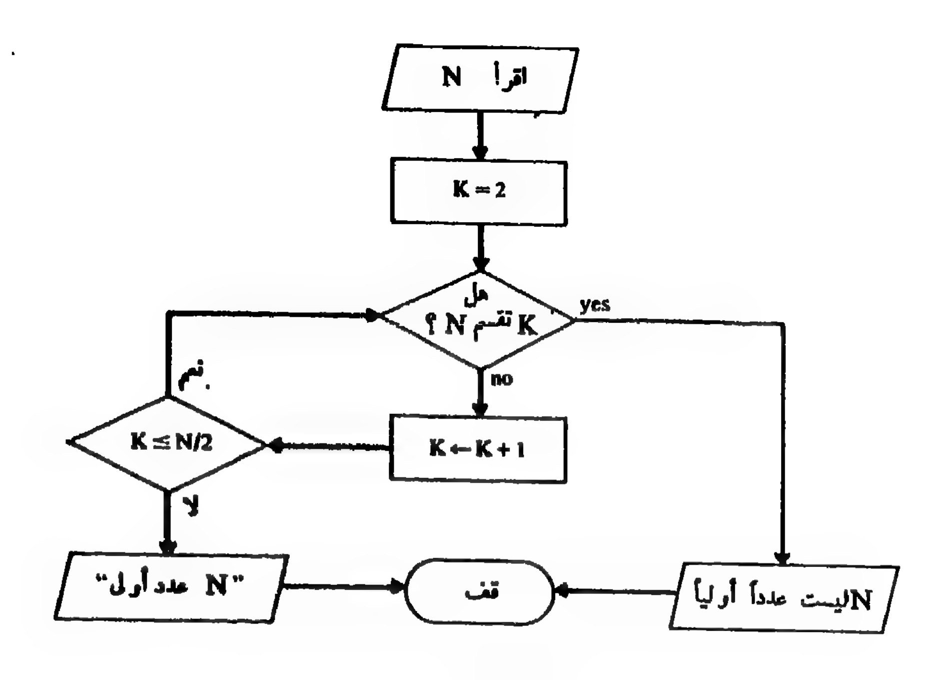
١ — ولد الأعداد الصحيحة 2 و 3 و . . . و N/2 ويمكن عمل هذا كما هو مبين فى شكل ٤ — ١٠ .



شكل ٤ — ١٠

٢ — اختبر إذا كانت N تقبل القسمة على كل عدد من الأعداد الصحيحة السابقة . إذا حدث ذلك بواسطة أي منها فإن N ليست عدداً أولياً ، وإلا فتكون N عدد أولى :

تظهر خريطة سير العمليات للنظام الحسابى (الخوارزم) فى شكل ٤ — ١١ .



شكل ٤ — ١١

قبل أن نترجم خريطة سير العمليات إلى فورتران ، يجب أن نكون قادرين على ترجمة الصندوق : Does k Devide N ? . ومع ذلك فنحن نعرف من مثال ٢ — ٩ أن N تقبل القسمة على K إذا كانت (N/K)*K لها نفس قيمة N ، وبذلك ، يكون لدينا البرنامج :

```
      READ(5, 20) N
   20 FORMAT(I5)
      K = 2
   30 IF(N/K*K.EQ.N) GO TO 70
      K = K + 1
      IF(K.LE.N/2) GO TO 30
      WRITE(6, 40) N
   40 FORMAT(1X, I5, 1X, 'IS A PRIME')
      STOP
   70 WRITE(6, 50) N
   50 FORMAT(1X, I5, 1X, 'IS NOT A PRIME')
      STOP
      END
```

٤ — ٧ جملة IF الحسابية

يوجد نوع آخر من جمل IF ضمن مجموعة الفورتران وتسمى جملة IF الحسابية . وعند مقارنتها بجملة IF المنطقية ، التي تختبر الحالة المنطقية لتعبير منطقي ، نجد أن جملة *IF* الحسابية تختبر إشارة تعبير حسابي . ويمكن أن ينتقل التحكم بعد ذلك إلى أماكن متعددة تبعاً لكونها موجبة أو صفر أو سالبة .

وفيما يلي شكل جملة IF الحسابية :

IF(*expr*) *l*, *m*, *n*

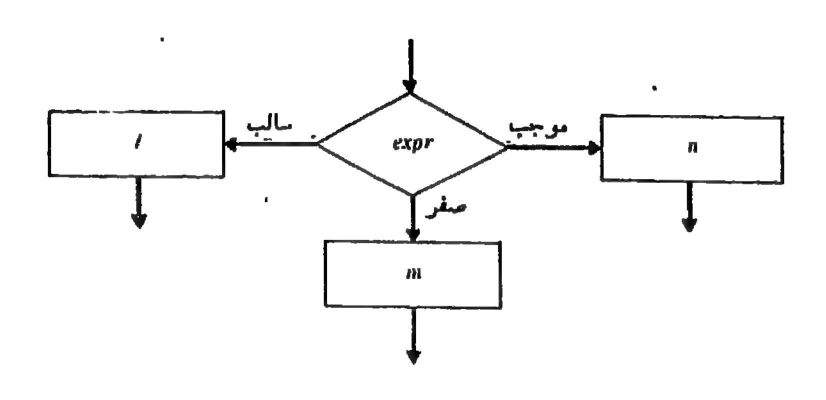
حيث ***expr*** هو تعبير حسابي و ***l***, ***m***, ***n*** أرقام جمل وتقول الجملة :

إذا كانت قيمة ***expr*** سالبة (<0) اذهب إلى الجملة رقم ***l***

وإذا كانت قيمة ***expr*** صفرا ($=0$) اذهب إلى الجملة رقم ***m***

إذا كانت قيمة ***expr*** موجبة (>0) إذهب إلى الجملة رقم ***n***

يجب أن تكون الجمل التي تحمل أرقام ***l, m, n*** جملا قابلة التنفيذ ، وليس من المطلوب أن تكون جميعها مختلفة ؛ بمعنى أن أي اثنين منهما أو جميعها يمكن أن تكون نفس الشيء . وخريطة سير العمليات المكافئة لجملة IF الحسابية موضحة في شكل ٤ — ١٢ .



شكل ٤ – ١٢

يمكن لأى جملة IF حسابية أن تنفذ بتكافؤ إذا استخدمنا واحدة جملة IF المنطقية مع جملة GO TO أو أكثر فعلى سبيل المثال :

(٤) IF(X – 2.0) 10, 20, 30 and IF(X – 2.0) 10, 10, 20

يكافئان على الترتيب :

```
IF(X.LT.2.0) GO TO 10              IF(X.LE.2.0) GO TO 10
IF(X.EQ.2.0) GO TO 20      and     GO TO 20
GO TO 30
```

وبالعكس ، فيمكن لأى جملة IF منطقية أن تنفذ بتكافؤ باستعمال جمل IF الحسابية (انظر مسألة ٤ – ٥) يجدر ملاحظة أن بعض المسائل المحددة تناسب بصورة طبيعية جملة IF الحسابية كما نرى فى المثال التالى :

مثال ٤ – ٦

(أ) افرض أن قسط التأمين الصحى الذى يخصم من مرتب موظف يحسب تبعاً للخطة التالية :

قسط = { 9.75 إذا كان أعزب ؛ 16.25 إذا كان متزوجاً ولا يعول ؛ 24.50 إذا كان متزوجاً ويعول }

افرض أن TYPE هى المتغير الصحيح الذى يشير إلى الحالة الاجتماعية بأكواد كما يلى :

1 أعزب ، 2 متزوج ولا يعول ، 3 متزوج ويعول . يعطى شكل ٤ – ١٣ خريطة سير العمليات لجزء من برنامج يخصم القسط من مرتب الموظف إذا افترضنا معرفة PAY و TYPE و ID الموظف . لاحظ أن TYPE — 2 تكون سالبة ، أو صفر ، أو موجبة تبعاً لقيمة TYPE إذا كانت 1 أو 2 أو 3 . فيما يلى ترجمة شكل ٤ – ١٣ إلى فورتران :

```
      REAL NET
      IF(TYPE - 2) 10, 20, 30
10    NET = PAY - 9.75
      GO TO 75
20    NET = PAY - 16.25
      GO TO 75
30    NET = PAY - 24.50
75    WRITE(6, 40) ID, NET
40    FORMAT(1X, I5, 3X, F12.2)
```

TYPE−2
سالب
صفر
موجب
NET←PAY−9.75
NET←PAY−16.25
NET←PAY−24.50
اطبع ID و NET

شكل ٤ – ١٢

(ب) أدرس المعادلة التربيعية :

$$ax^2 + bx + c = 0$$

حيث a و b و c أرقام حقيقية و $0 \neq a$ ستكون هناك جذور حقيقية فقط عندما يكون المميز $D = b^2 - 4ac$ غير سالب وتعطى الجذور بواسطة :

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

لاحظ أنه إذا كان المميز $D = 0$ فإننا نحصل على الجذر المتعدد $b/2a$ — اكتب برنامج فورتران يحسب الجذور الحقيقية لمجموعة معطاه من المعاملات A ، B ، C

```
C
C     COMPUTE THE REAL ROOTS OF A QUADRATIC EQUATION WHERE A
C          IS NOT ZERO
C
      READ(5, 11) A, B, C
11    FORMAT(3F10.2)
      WRITE(6, 12) A, B, C
12    FORMAT('0', 'THE COEFFICIENTS ARE', 3(2X, F10.2))
      D = B**2 - 4.0*A*C
C
C     TEST THE DISCRIMINANT
C
      IF(D) 22, 33, 44
22    WRITE(6, 13)
13    FORMAT(1X, 'THERE ARE NO REAL ROOTS')
      GO TO 10
33    ROOT = -B/(2.0*A)
      WRITE(6, 14) ROOT, ROOT
14    FORMAT(1X, 'THERE ARE TWO IDENTICAL ROOTS', 2(3X, F10.2))
      GO TO 10
44    ROOT1 = (-B + SQRT(D))/(2.0*A)
      ROOT2 = (-B - SQRT(D))/(2.0*A)
      WRITE(6, 15) ROOT1, ROOT2
15    FORMAT(1X, 'THERE ARE TWO DISTINCT ROOTS', 2(3X, F10.2))
10    STOP
      END
```

٤ — ٨ جملة GO TO المحسوبة

جملة GO TO المحسوبة هي جملة أخرى من جمل التحكم المشروط . في الحقيقة ، فهذه الجملة تسمح بأي عدد من الاختيارات التفرع . ويعتمد قرار التفرع المعين على قيمة متغير صحيح يظهر في الجملة .

وفيما يلي الشكل العام لجملة GO TO المحسوبة :

GO TO ($n_1, n_2, \ldots, n_k$), J

حيث n_1 ، $n_2 \ldots n_k$ ثوابت صحيحة بدون إشارة وتمثل أرقام جمل و J هي متغير صحيح بدون دليل . لاحظ أن أرقام الجمل تفصل عن بعضها بواسطة فصلات وتحاط بأقواس ، والأقواس متبوعة بفصلة أخرى ثم بعد ذلك اسم المتغير J .

وجملة GO TO المحسوبة السابقة لها المعنى التالي : عند وقت التنفيذ « تختبر قيمة المتغير الصحيح J أولا . إذا كانت قيمة J هي 1 ، انتقل التحكم إلى الجملة التي تحمل الرقم n_1 (أي ، عندئذ تنفذ الجملة رقم n_1) ، وإذا كانت قيمة J هي 2 عندئذ تنفذ الجملة التي تحمل الرقم n_2 وهكذا ، وأنه لغاية في الأهمية أن تقع قيمة J بين 1 و k أثناء وقت التنفيذ وإلا لا نستطيع أن نتنبأ بالإجابة . في الحقيقة ، أن بعض المترجمات لا تعطي حتى رسالة خطأ .

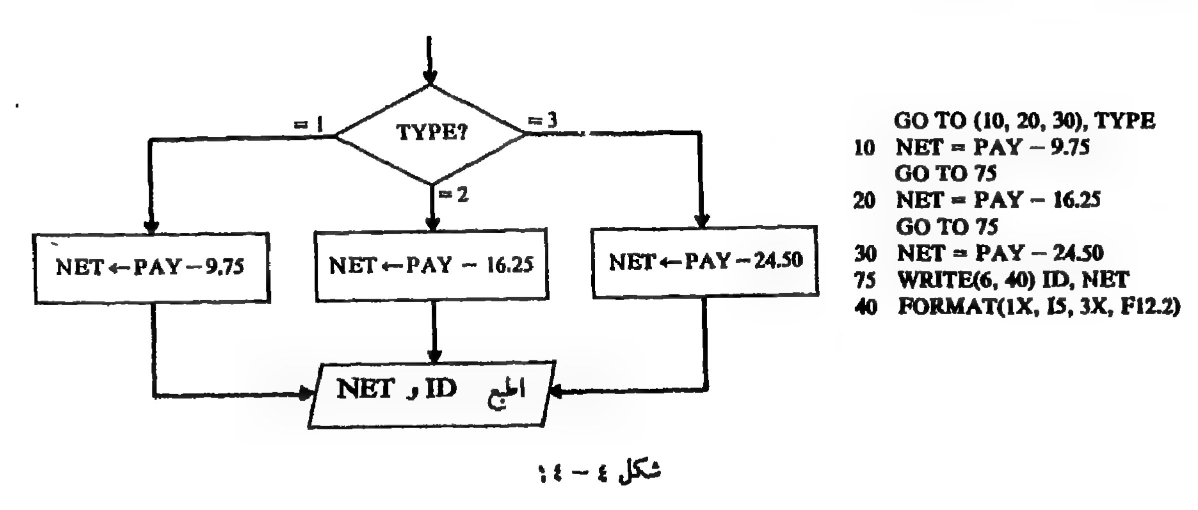
افرض على سبيل المثال ، أنه تم تنفيذ الجملة التالية :

GO TO (10, 15, 70, 22, 15), NEW

ينتقل التحكم إلى الجملة التي تحمل الرقم 10 أو 15 أو 70 أو 22 أو 15 تبعاً للقيمة التي تأخذها NEW 1 أو 2 أو 3 أو 4 أو 5 على الترتيب . لاحظ أن أرقام الجمل ليس من المطلوب أن تكون مختلفة . ولكن نؤكد أنه عند وقت التنفيذ يجب أن تكون NEW موجبة ولا تتعدى 5 .

٧ —

ادرس مثـال ٤ — ٤ حيث يخصم قسط التأمين الصحي تبعاً لكون الموظف أعزب (TYPE 1) أو متزوجاً ولا يعول (TYPE 2) أو متزوجاً ويعول (TYPE 3) يظهر في شكل ٤ — ١٤ خريطة سير العمليات وجزءاً من برنامج الفورتران باستعمال جملة GO TO المحسوبة .




```
      GO TO (10, 20, 30), TYPE
10    NET = PAY - 9.75
      GO TO 75
20    NET = PAY - 16.25
      GO TO 75
30    NET = PAY - 24.50
75    WRITE(6, 40) ID, NET
40    FORMAT(1X, I5, 3X, F12.2)
```

شكل ٤ — ١٤

يمكن أن نرى بسهولة أن جملة GO TO المحسوبة لا تعطينا أى تسهيلات حماية إضافية . أى أننا يمكن أن ننظم سلسلة من جمل IF المنطقية لإنجاز نفس نتيجة جملة واحدة من جمل GO TO المحسوبة . على سبيل المثال :

```
IF(MM.EQ.1) GO TO 99
IF(MM.EQ.2) GO TO 88
IF(MM.EQ.3) GO TO 101
IF(MM.EQ.4) GO TO 23
```

تكافئ

```
GO TO (99, 88, 101, 23), MM
```

ومع ذلك فمجموعة جمل IF المنطقية أقل كفاءة من جملة واحدة من جمل GO TO المحسوبة ، والتى تتطلب خطوة واحدة فقط لنقل التحكم إلى المكان المطلوب . وكما نرى أنها تحتاج إلى كتابة أقل .

ملاحظة : هناك نوع آخر من جمل التحكم المشروط والتى تسمح بالتفرع المتعدد ، وهى جملة GO TO المخصصة . وستناقش هذه الجملة مع جملة ASSIGN المصاحبة فى الفصل الحادى عشر .

٤ ــ ٩ الخوارزميات (نظم الحساب)

سنناقش برنامجين فى هذا القسم . والغرض هو توضيح المراحل المختلفة المتعلقة بكتابة البرنامج بدءاً من وضع النظام الحسابى (الخوارزم) ثم تهذيبه ، إلى تنفيذه ، ولن تتم ترجمة خرائط سير العمليات النهائية هنا إلى فورتران ، بل ستترك كتمرينات (مسائل تكميلية) للقارىء .

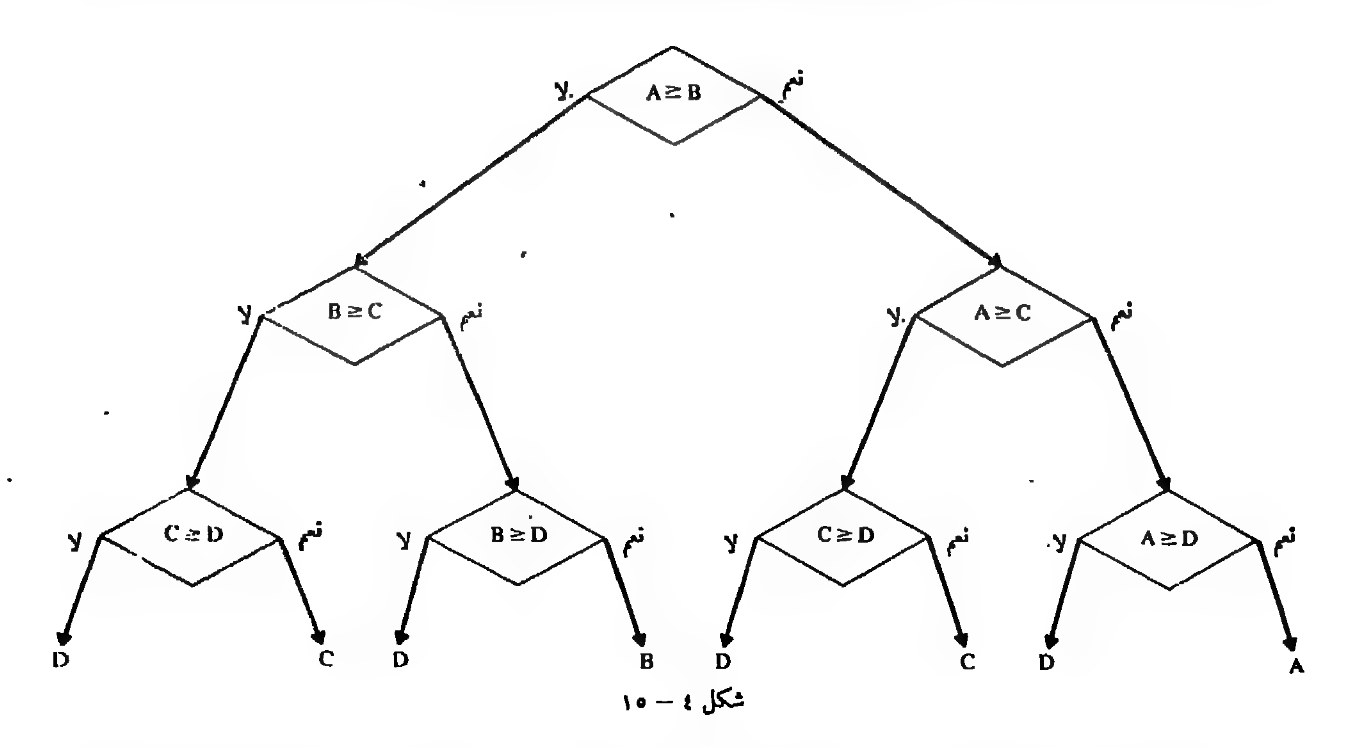
افرض أنه تم اعطاء مجموعة من أربع بطاقات مثقب على كل بطاقة عدد . والمطلوب إيجاد أكبر عدد من هذه الأعداد . ولما كان هناك أربعة أعداد فقط . دعنا نقرأها داخل الذاكرة ونطلق عليها A و B و C و D . سيعطينا النظام الحسابى (الخوارزم) التالى أكبر عدد بفرض أن A و B و C و D فى الذاكرة .

١ ــ قارن A و B وأوجد أيهما أكبر .

٢ ــ قارن C مع الناتج من 1 وأوجد أيهما أكبر .

٣ ــ قارن D مع الناتج من 2 وأوجد أيهما أكبر .

وخريطة سير العمليات فى شكل ٤ ــ ١٥ توضح أحد الأساليب المباشرة والبديهية للنظام الحسابى (الخوارزم) السابق .



شكل ٤ — ١٥

لاحظ أن هناك سبعة ($2^2 + 2^1 + 1 = 7$) صناديق اتخاذ قرار فى شكل ٤ — ١٥ وفى الحقيقة ، العدد الكلى لصناديق اتخاذ القرار فى مثل هذا النظام الحسابى (الخوارزم) تنمو أسيا مع عدد العناصر ، بالتحديد لو كان هناك خمسة أرقام ، سيكون هناك ($2^3 + 7 = 15$) خمسة عشر صندوق لاتخاذ قرار ، ولو كان هناك ستة أرقام ، سيكون هناك ($2^4 + 15 = 31$) إحدى وثلاثون صندوق لاتخاذ قرار ، وهكذا من الواضح أن هذا النظام الحسابى (الخوارزم) ليس مقنعاً عندما يكون عدد العناصر كبيرا .

نهذب نظامنا الحسابى (الخوارزم) قليلا بجعل LAR تشير إلى القيمة الكبرى ونستعمل الخطوات الأربع التالية :

(٠) خصص A إلى LAR أى ضع LAR = A

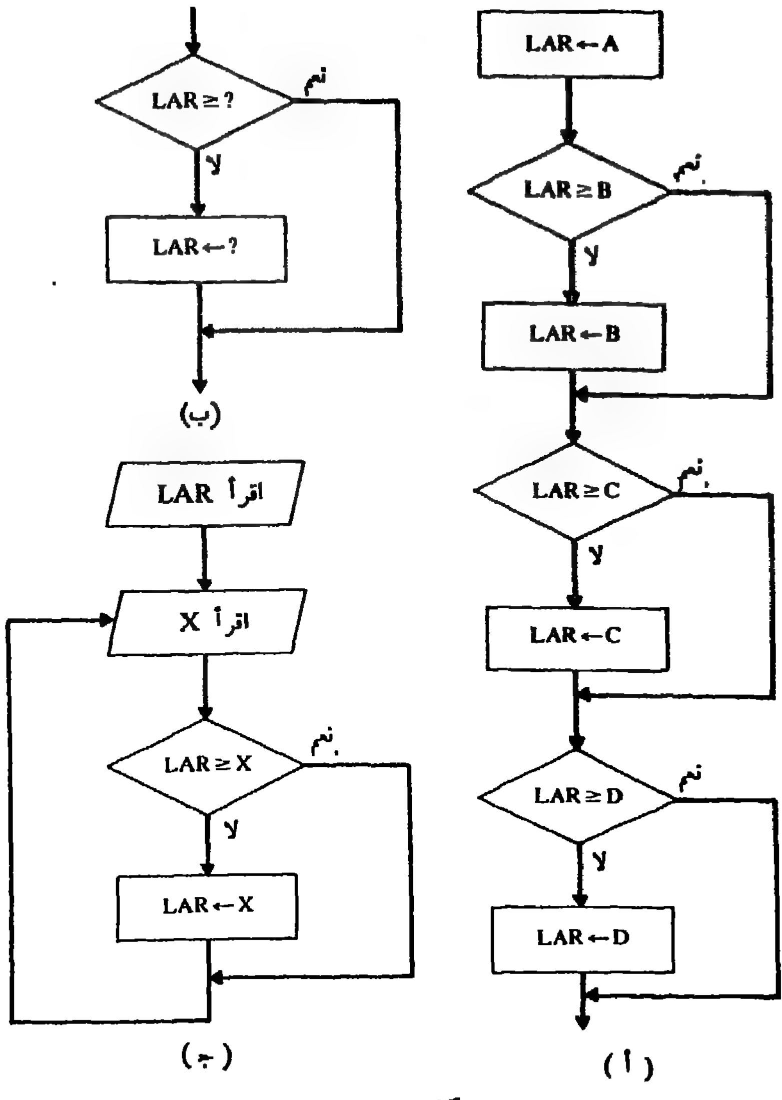
(١) قارن LAR و B واحتفظ بالأكبر فى LAR

(٢) قارن LAR و C واحتفظ بالأكبر فى LAR

(٣) قارن LAR و D واحتفظ بالأكبر فى LAR

خريطة سير العمليات لهذا النظام الحسابى (الخوارزم) معطاه فى شكل ٤ — ١٦ (أ) لاحظ أن عدد صناديق اتخاذ قرار ثلاثة ، وتنمو خطياً مع عدد العناصر . أى ، بالنسبة للارقام الخمسة ، يكون هناك أربعة صناديق اتخاذ قرار ، ولعدد k من الأرقام ، سيكون هناك $k - 1$ صناديق اتخاذ قرار . واضح ، أن هذا النظام الحسابى (الخوارزم) أكثر تقدماً من النظام الحسابى (الخوارزم) السابق .

من الأشياء الهامة التى تلفت النظر فى خريطة سير العمليات بالشكل ٤ — ١٦ (أ) هو تكرار مجموعة معينة من الصناديق وسير العمليات مرة بعد أخرى . وهذه المجموعة مبينة فى شكل ٤ — ١٦ (ب) حيث ؟ تمثل عنصراً اختيارياً (يخالف الأول) ، وبالتالى لو استطعنا أن نضمن أن قيمة ؟ تتغير بطريقة صحيحة ، سيكن حينئذ أن نجمع الصناديق فى حلقة تكرارية . ويتم هذا التغيير بقراءة القيم واحدة تلو الأخرى باستعمال نفس الاسم (مكان التخزين) فى كل مرة والحلقة التكرارية اللانهائية الناتجة عن ذلك مبينة فى شكل ٤ — ١٦ (ج) .



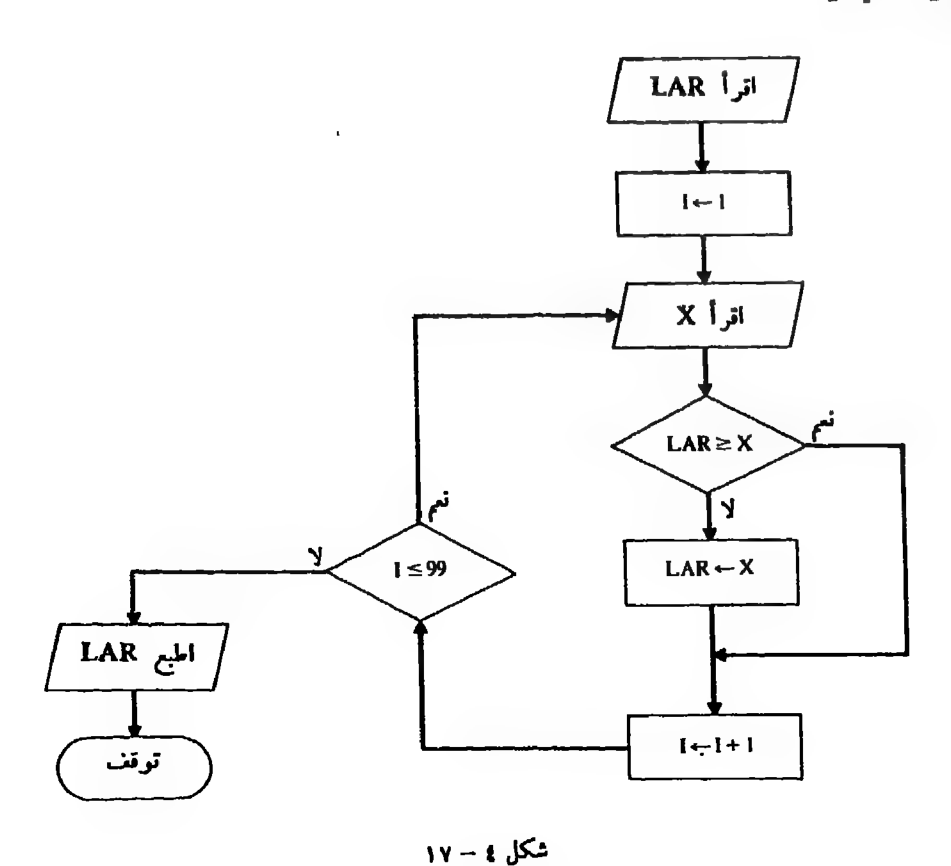
شكل ٤ — ١٦

والآن افرض أن المجموعة المعطاه 100 بطاقة . فى هذه الحالة تظهر ، خريطة سير العمليات لإيجاد الرقم الأكبر كما فى شكل ٤ — ١٧ (ولما كان قد خصص الرقم الأول لـ LAR فى البداية فإن العملية تتكرر 99 = 1 — 100 مرة فقط) ويظهر برنامج الفورتران لمسألة أعم من ذلك قليلا فى مسألة ٤ — ١٣ .

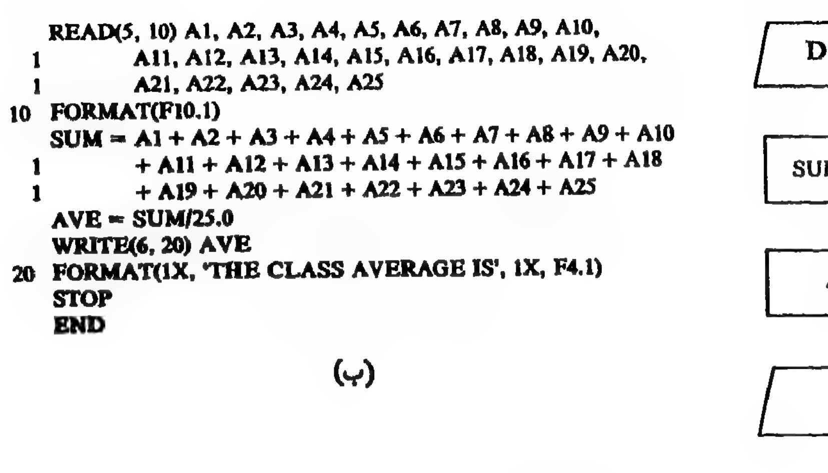
لندرس الآن مثالا آخر . افرض مجموعة من أربع بطاقات ومثقب على كل بطاقة عدد ما . المطلوب إيجاد المتوسط (المتوسط الحسابى) لهذه الأعداد . والطريقة المباشرة هى أن تقرأ هذه الأعداد داخل أماكن تخزين ، ولتكن A و B و C و D ثم نقسم مجموعها على 4 وتظهر خريطة سير العمليات فى شكل ٤ — ١٨ (أ)

والآن نفرض أن مجموعة البطاقات هى 25 وليست 4 بطاقات . فباستخدام النهج السابق ، سيتطلب ذلك قراءة 25 متغيراً مختلفاً ، ليكن A_1 ، A_2 .... A_{25} (الأعداد 1 ، 2 ، ...25 فى أسماء المتغيرات لا تلعب أى دور سوى كونها رموزاً مختلفة) . ويعطى الشكل ٤ — ١٨ (ب) برنامج الفورتران الذى سيحسب متوسط 25 عدداً . لاحظ أن كلا من جملة READ والجملة التى تحسب SUM طويلة جداً ومن السهل حدوث أخطاء بها . كذلك يلاحظ أنه إذا كان هناك مائة رقم فسيتعذر تماماً

التحكم فى الحل بهذا الأسلوب . وعلاوة على ذلك ، سنضطر أيضاً لإعادة كتابة البرنامج . ويتضح لنا بالتالى أنه من الأفضل استخدام نظام حسابى (خوارزم) معين بحيث يمكن استخدام نفس البرنامج (مع احتمال تغيير طفيف) لحساب مجموع 4 أعداد أو 25 أو 100 عدد . وسنناقش هذا فيما يلى .

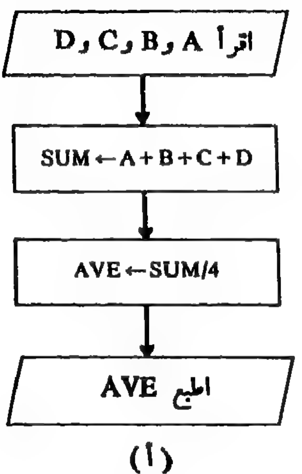


شكل ٤ – ١٧



```
      READ(5, 10) A1, A2, A3, A4, A5, A6, A7, A8, A9, A10,
     1          A11, A12, A13, A14, A15, A16, A17, A18, A19, A20,
     1          A21, A22, A23, A24, A25
   10 FORMAT(F10.1)
      SUM = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + A10
     1          + A11 + A12 + A13 + A14 + A15 + A16 + A17 + A18
     1          + A19 + A20 + A21 + A22 + A23 + A24 + A25
      AVE = SUM/25.0
      WRITE(6, 20) AVE
   20 FORMAT(1X, 'THE CLASS AVERAGE IS', 1X, F4.1)
      STOP
      END
```

(ب)



(أ)

شكل ٤ – ١٨

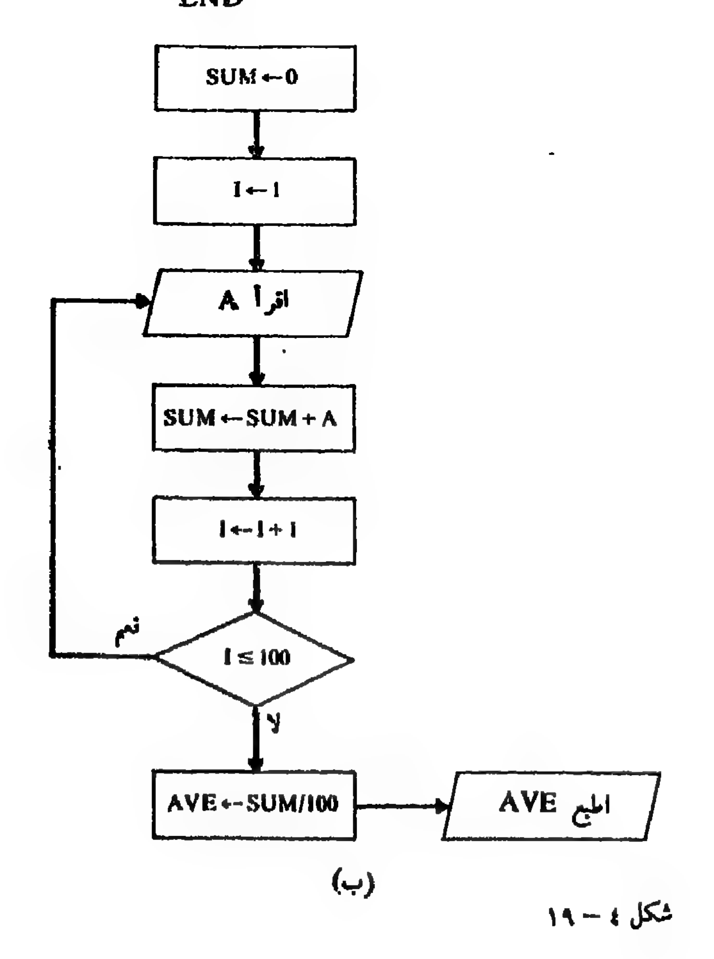
٨ — البرمجة بلغة الفورتران

لحساب ... + $A_1 + A_2 + A_3 + A_4$ يمكن أن نعتبرها

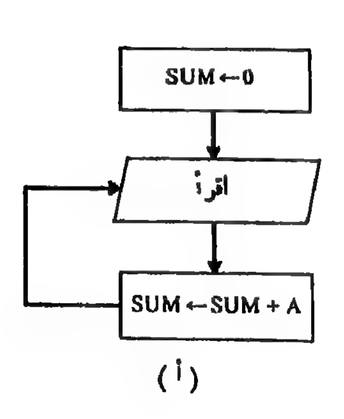
$$((((A1) + A2) + A3) + A4) + \cdots$$

أى يمكن أن نبدأ بـ SUM = 0 ثم يتم جمع هذه الأعداد واحدة تلو الأخرى إلى محتويات SUM . فى الواقع ، فإننا يمكن أن نجمعها واحدة تلو الأخرى من البداية ، أى عند قراءتها وتخزينها داخل الذاكرة ، كما هو موضح فى الحلقة التكرارية اللانهائية من الشكل ٤ – ١٩ (أ) . والآن إذا كان هناك مجموعة من 100 بطاقة ومثقب فى كل منها عدد واحد ، فيمكن أن يجهز عداد I ليعد عدد المرات التى تنفذ فيها الحلقة التكرارية . وبذا يتم ايقاف العملية بعد تنفيذ جمع المائة عدد . وهذا يعطى خريطة سير العمليات التى فى الشكل ٤ – ١٩ (ب) وفيما يلى ما يقابله بالفورتران .

```
      SUM = 0.0
      I = 1
50    READ(5, 10) A
10    FORMAT(F10.2)
      SUM = SUM + A
      I = I + 1
      IF(I.LE.100) GO TO 50
      AVE = SUM/100.0
      WRITE(6, 20) AVE
20    FORMAT(1X, 'THE AVERAGE IS', 1X, F4.1)
      STOP
      END
```



(ب)



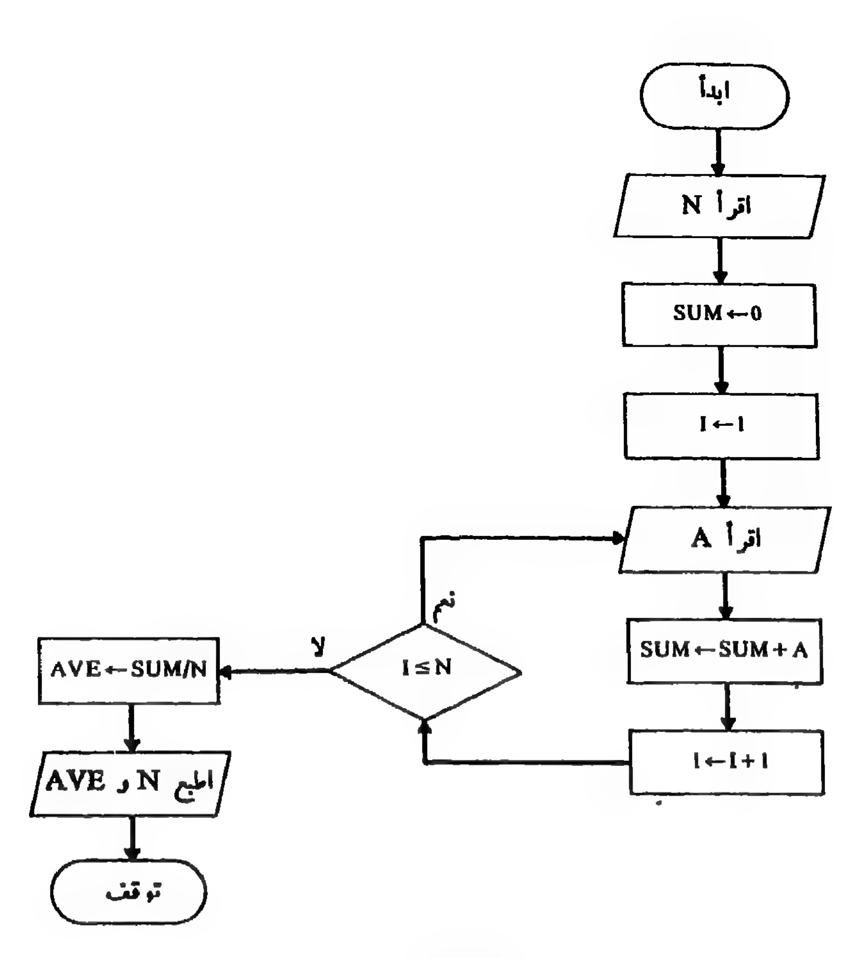
(أ)

شكل ٤ – ١٩

٤ ـــ ١٠ بطاقة المقدمة والبطاقة الخلفية

البرامج التي تم تطويرها في قسم ٤ – ٩ بها عيب واحد ، فجميعها سوف تعمل كما يجب إذا كان هناك مجموعة من 100 بطاقة فقط . يجب أن يعدل البرنامج إذا كانت المجموعة المعطاة تتكون من 76 بطاقة ، مع أن النظام الحسابي (الخوارزم) نفسه لا يتغير ومن الواضح أن المطلوب هو كتابة برنامج لا يعتمد على عدد البطاقات الموجودة في مجموعة البيانات ، أي أن المطلوب كتابة برنامج يعمل لأى عدد من بطاقات البيانات سواء كانت 4 أو 100 أو 76 . وسوف نناقش طريقتين يمكننا بهما أن ننجز ذلك : (١) استخدام بطاقة المقدمة و (٢) استخدام البطاقة الخلفية .

في خريطة سير العمليات في شكل ٤ – ١٩ (ب) ، نجد أن الحلقة التكرارية تتكرر 100 مرة وذلك لإيجاد مجموع 100 قيمة . افرض ، أننا نريد أن نكرر الحلقة التكرارية لعدد متغير من المرات (تبعاً لمسألة معينة) وعليه يجب أن نسمى هذا العدد باسم متغير وليكن N . لتنفيذ البرنامج ، مع استخدام المتغير N بدلا من العدد 100 يجب أن تعرف قيمة N عند وقت التنفيذ . ويمكن أن يتم ذلك بتثقيب عدد بطاقات المجموعة على بطاقة تسمى **بطاقة المقدمة** وتوضع أعلى مجموعة البيانات . وبقراءة هذا العدد يتم تعريف N وخريطة سير العمليات لهذا البرنامج معطاة في شكل ٤ – ٢٠ .



شكل ٤ – ٢٠

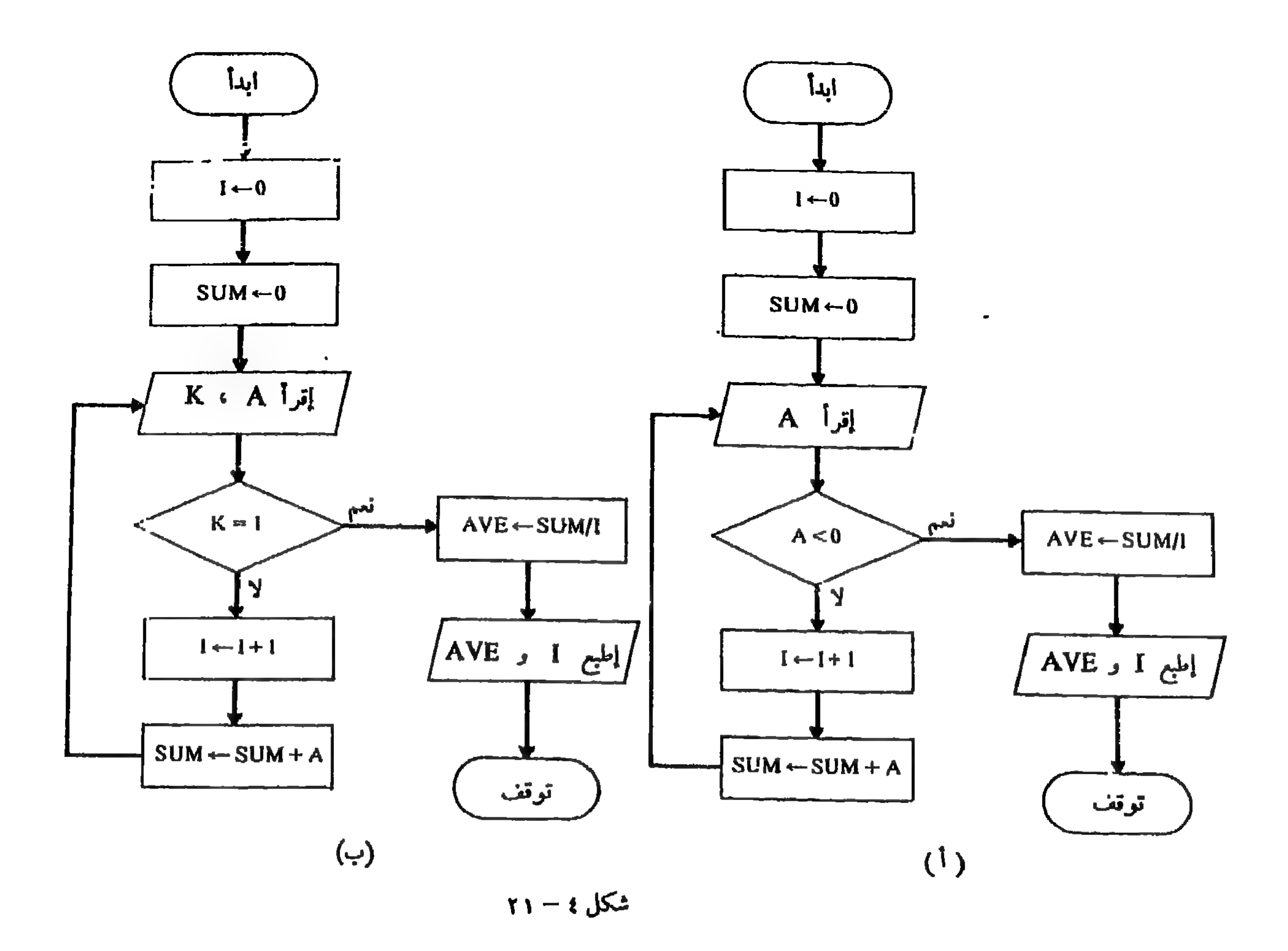
يصعب أحياناً استخدام بطاقة المقدمة فمثلا إذا كان المطلوب حساب متوسط درجات الامتحان لعدد من الطلبة يتراوح ما بين 200 و 300 طالب فلكى نعد بطاقة المقدمة ، يجب أولا أن يتم تحديد عدد بطاقات البيانات فى المجموعة . وبما أن عدد الطلبة يتراوح ما بين 200 و 300 طالب ، فعدد البطاقات سيتراوح بين 200 و 300 بطاقة . ويستحسن بالطبع أن يقوم الحاسب بعملية العد . ويمكن أن يتم هذا باستخدام البطاقة الخلفية ، التى نناقشها فيما يلى .

افرض أننا سنحسب متوسط درجات امتحان كما سبق . ولما كان العدد الذى يمثل درجة الأمتحان يقع ما بين صفر و 100 فقط ، فنستطيع أن نضع فى آخر المجموعة بطاقة إضافية تحتوى على رقم خارج هذا النطاق : كاستخدام عدد سالب مثلا . وتسمى هذه البطاقة البطاقة الخلفية . وحين يقابل الحاسب عدداً سالباً ، يعطى إشارة بأن مجموعة البيانات قد إنتهت وخريطة سير العمليات لمثل هذا البرنامج تظهر فى شكل ٤ — ٢١ (أ) .

ملاحظة : لاحظ أنه فى هذا البرنامج أعطى العداد قيمة ابتدائية صفراً . والسبب فى ذلك أننا نريد أن نعرف عدد الأعداد الموجبة فقط ، إذ أننا لا نعرف أن العدد موجب إلا بعد أن يتم اختياره . وبذلك ، فقيمة I بعد تنفيذ I = I + 1 تحدد آخر بطاقة تمت قراءتها وليس البطاقة الجارى قراءتها .

يعمل بالأسلوب السابق فقط عندما نعرف مدى الأعداد ، إلا أنه بعد إجراء تغيير فيه يمكن استخدامه حتى إن لم تكن هناك أى معرفة عن البيانات التى سيتناولها ، يمكن أن تقرأ كأصفار (فى الحقول الرقية) فالحاسب يشعر بنهاية مجموعة البيانات حين تأخذ K القيمة 1 وتظهر خريطة سير العمليات لهذا البرنامج البديل فى شكل ٤ — ٢١ (ب) .

أخيراً نشير إلى أنه فى بعض الاحيان يطلق على بطاقات المقدمة والبطاقات الخلفية البطاقات الحارسة .



شكل ٤ — ٢١

مســـائل محلولة

جمل IF

٤ – ١ اكتب الجمل الآتية بالفورتران

(أ) If X > Y, stop.
(ب) If J ≠ K, go to the statement labeled 31.
(جـ) If $A^2 \leq B + C$, go to the statement labeled 41.
(د) If $A - B \geq X^3$, stop.

(أ) IF(X.GT.Y) STOP
(ب) IF(J.NE.K) GO TO 31
(جـ) IF(A**2.LE.B + C) GO TO 41
(د) IF(A – B.GE.X**3) STOP

٤ – ٢ إفرض أنه تم تعريف X و Y. اكتب جملة فورتران أو جزءاً من برنامج فورتران الذى (أ) ينقل التحكم إلى الجملة رقم 41 إذا كانت $X^2 \leq Y$ وينقل التحكم إلى الجملة رقم 42 إذا كانت غير ذلك ، (ب) يضع K = 0 إذا كانت $X + Y > 100$ ويضع K = 1 إذا كانت غير ذلك .

نفذ ماسبق بطريقتين ، مرة بجملة IF الحسابية ، ومرة أخرى بجملة IF المنطقية .

(أ) (١) لاحظ أن الشرط $X^2 \leq Y$ يتحقق إذا كانت قيمة $X^2 - Y$ سالبة أو صفراً فقط .

```
      IF(X**2 – Y) 41, 41, 42
```

(٢)

```
      IF(X**2.LE.Y) GO TO 41
42    ......................
```

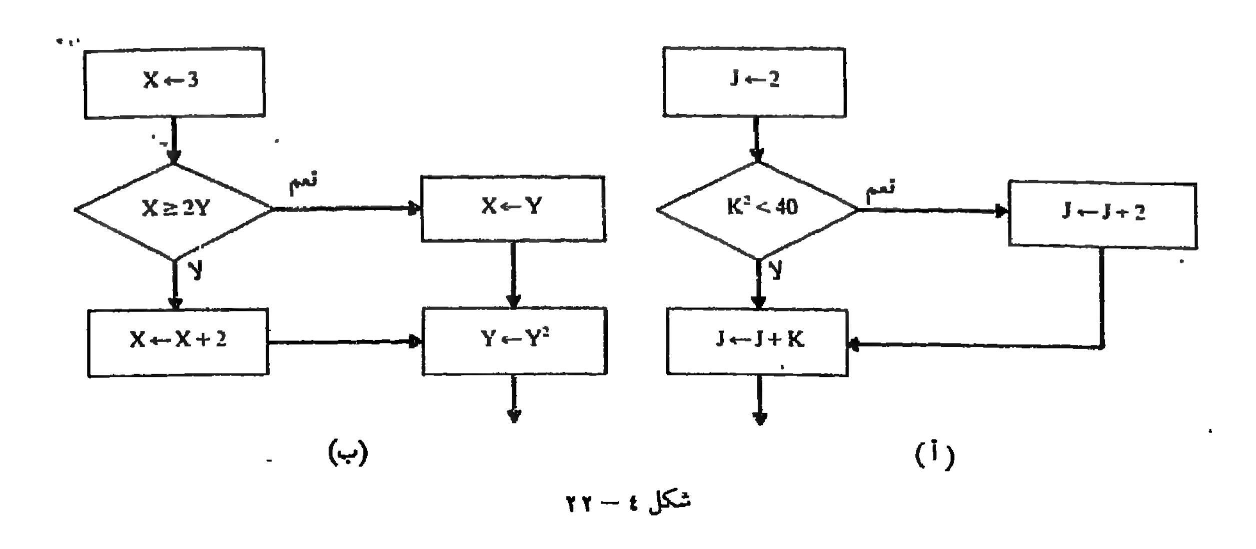
(ب) (١) لاحظ أن الشرط $X + Y > 100$ يتحقق فقط إذا كانت قيمة $X + Y - 100$ موجبة .

```
      IF(X + Y – 100.0) 20, 20, 10
10    K = 0
      GO TO 30
20    K = 1
30    ****************************
```

(٢)

```
      IF(X + Y.LE.100.0) GO TO 20
      K = 0
      GO TO 30
20    K = 1
30    ****************************
```

٤ — ٣ ترجم خريطتى سير العمليات فى الشكل ٤ — ٢٢ إلى أجزاء برنامج فورتران .



شكل ٤ — ٢٢

(أ) الخريطة تكافىء IF... THEN... كما سبق وأن ناقشناها فى حالة (١) من قسم ٤ — ٥ .

```
J = 2
IF(K**2.LT.40) J = J + 2
J = J + K
```

(ب) الخريطة تكافىء IF...THEN..ELSE كما سبق وأن ناقشناها فى حالة (٢) من قسم ٤ — ٥ .

```
   X = 3.0
   IF(X.GE.2*Y) GO TO 10
   X = X + 2.0
   GO TO 20
10 X = Y
20 Y = Y**2
```

٤ — ٤ إفرض أنه تم تخزين T_1 و T_2 و T_3 فى الذاكرة وترمز هذه المتغيرات إلى درجات اختبار . اكتب جزءاً من برنامج الفورتران الذى يجد ويطبع عدد الدرجات التى تساوى أو تزيد عن 90 (مسألة أكثر شمولا معطاه فى ٤ — ١٤) .

بجعل N ترمز إلى عدد درجات أكبر من أو تساوى 90 (لذا ، N يجب أن تساوى 0 أو 1 أو 2 أو 3)

```
   N = 0
   IF(T1.GE.90.0) N = N + 1
   IF(T2.GE.90.0) N = N + 1
   IF(T3.GE.90.0) N = N + 1
   WRITE(6, 10) N
10 FORMAT(1X, I5)
```

٤ — ٥ اكتب جزءاً من برنامج فورتران باستعمال جملة IF الحسابية التي تكافيء كلا مما يأتي (جمل S و T هنا قابلة التنفيذ)

(أ) IF(A.LT.B) S
T

(ب) IF(J.NE.K) S
T

(أ) لاحظ أن الشرط (A < B) يتحقق فقط إذا كانت (A—B) سالبة

```
   IF(A - B) 10, 20, 20
10 S
20 T
```

(ب) لاحظ أن الشرط (J ≠ K) يتحقق فقط إذا كانت (J — K ≠ 0) .

```
   IF(J - K) 10, 20, 10
10 S
20 T
```

٤ — ٦ إفرض أن J تحتوى على 5 و K تحتوى على 10 . اوجد القيمة النهائية لـ J بعد كل جزء من برامج الفورتران التالية :

(أ)
```
   IF(3*J.LT.K) J = J + 2
   J = J + 3
```

(ب)
```
   IF(2*J.EQ.K) J = J + 2
   J = J + 3
```

(ﺟ)
```
   IF(K - J) 10, 20, 10
10 J = K
20 J = J + 1
```

(د)
```
   IF(4*J - 2*K) 10, 20, 20
10 J = K
20 J = J + 1
```

(ﻫ)
```
   IF(2*K.LE.3*J) GO TO 50
   J = J + 1
   GO TO 60
50 J = K
60 J = J + K
```

(و)
```
   IF(K.GT.J) GO TO 50
   J = J + 1
   GO TO 60
50 J = K
60 J = J + K
```

(أ) حيث أن 3J < K غير صحيحة فإن J = J + 3 هى التي تنفذ ، لذا فإن قيمة J النهائية هى 8 .

(ب) حيث أن 2J = K صحيحة فإن J = J + 2 تنفذ أولا لتعطى J = 7 . ثم تنفذ بعد ذلك J = J + 3 ؛ ومن ثم تكون قيمة J النهائية هى 10.

(ﺟ) حيث أن K — J موجبة فتنفذ J = K أولا لتعطى J = 10. ثم تنفذ بعد ذلك J = J + 1 التي تعطى قيمة نهائية J = 11.

(د) حيث أن 4J — 2K هى صفر فتنفذ فقط J = J + 1 ، ومن ثم تكون قيمة J النهائية هى 6 .

(ﻫ) حيث أن 2K ≤ 3J غير صحيحة ، فتنفذ أولا J = J + 1 لتعطى J = 6 . ثم تنفذ بعد ذلك J = J + K ، والتي تعطى قيمة نهائية J = 16 .

(و) حيث أن K > J صحيحة فتنفذ أولا J = K لتعطى J = 10. ثم تنفذ بعد ذلك J = J + K التي تعطى قيمة نهائية J = 20 .

جملة GO TO المحسوبة

٤ — ٧ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل GO TO المحسوبة .

(أ) GO TO (5, 8, 4) MARK
(ب) GO TO (5, 22, 22, 57), KKK
(جـ) GO TO (2, 84, 578), ERIK
(د) GO TO (5, 76, 0, 24), J

(أ) يجب أن تكون هناك فصلة قبل MARK

(ب) لاتوجد أخطاء .

(جـ) يجب أن نكون ERIK متغيراً صحيحاً .

(د) لايمكن أن يكون الصفر رقم جملة .

٤ — ٨ أوجد رقم الجملة التى ينتقل إليها التحكم بعد كل جزء من برنامج الفورتران .

(أ)

```
MARK = 3
GO TO (23, 47, 16, 94), MARK
```

(ب)

```
J = 2
J = J + 2
GO TO (23, 47, 16, 94), J
```

(جـ)

```
K = 1
K = K + 3
GO TO (23, 16, 94), K
```

(أ) حيث أن MARK = 3 فإن التحكم ينتقل إلى الجملة رقم 16 وهو الرقم الثالث فى القائمة .

(ب) حيث أن J = 4 بعد تنفيذ جملة IF المحسوبة ، لذا ينتقل التحكم إلى الجملة رقم 94 .

(جـ) حيث أن K = 4 بعد تنفيذ جملة GO TO المحسوبة ، وحيث أن القائمة تحتوى على ثلاثة أرقام فقط ، ومن ثم ، ستعطى رسالة خطأ .

البرامج

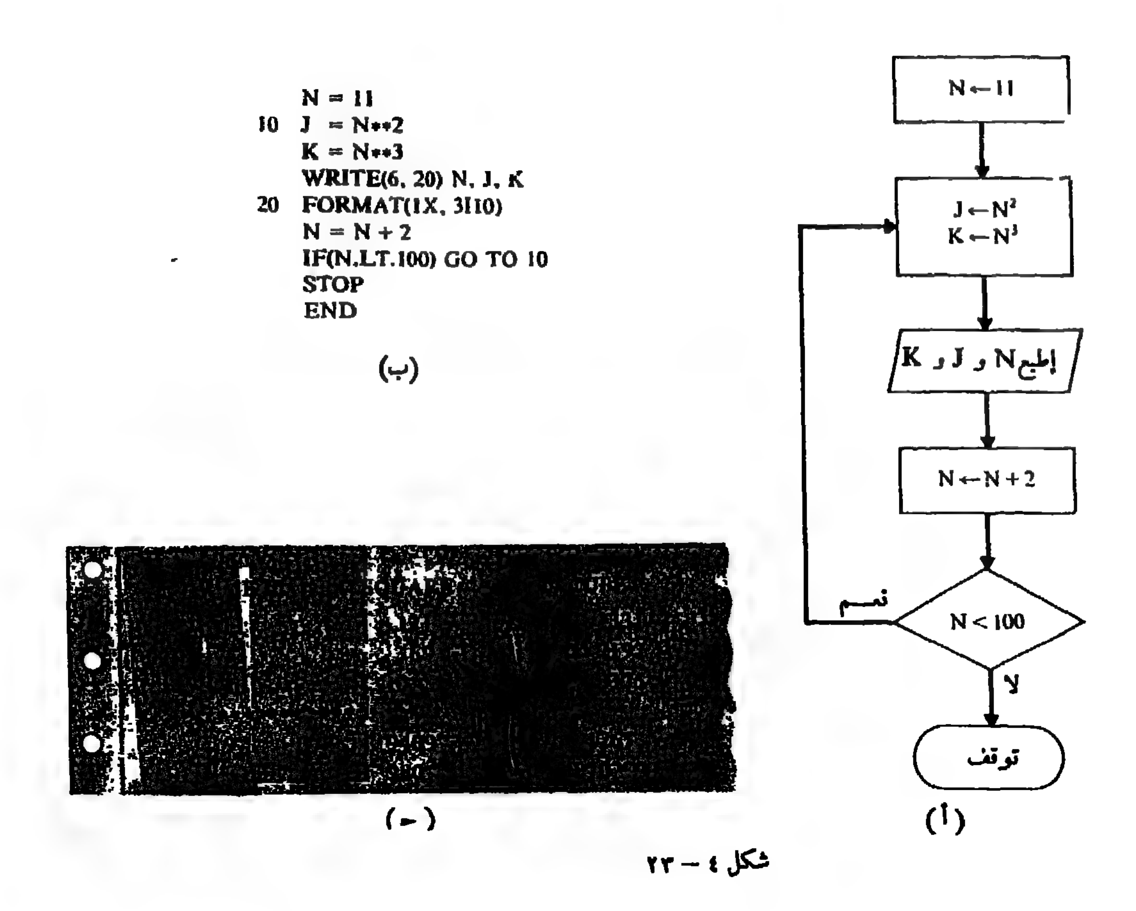
٤ — ٩ ارسم خريطة سير العمليات وأكتب البرنامج الذى يطبع كل عدد فردى من رقمين N ومربعة N^2 وتكعيبه N^3 بحيث تظهر قيم N المختلفة على أسطر مختلفة . يجب أن يكون البرنامج مكتوباً بحيث يكون للعمود N العنوان NUMBER ، وللعمود N^2 العنوان SQUARE وللعمود N^3 العنوان CUBE .

نجعل N = 11 و $J = N^2$ و $K = N^3$ نختبر لنرى إذا كانت N < 100 وذلك بعد زيادة N بمقدار 2 . خريطة سير العمليات معطاة فى الشكل ٤ — ٢٣ (أ) وما يقابلها بالفورتران فى شكل ٤ — ٢٣ (ب) .

لو أضفنا الجملتين التاليتين فى بداية البرنامج

```
  WRITE(6, 5)
5 FORMAT(7X, 'NUMBER', 3X, 'SQUARE', 5X, 'CUBE')
```

حينئذ يكون الخرج موضحاً كما فى الشكل ٤ — ٢٣ (جـ) .




```
      N = 11
10    J = N**2
      K = N**3
      WRITE(6, 20) N, J, K
20    FORMAT(1X, 3I10)
      N = N + 2
      IF(N.LT.100) GO TO 10
      STOP
      END
```

شكل ٤ — ٢٣

٤ — ١٠ ارسم خريطة سير العمليات واكتب برنامج فورتران يحسب التالى (إلى خمسة أرقام عشرية)

(أ) $\frac{1}{1}+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{21}$ (ب) $\frac{2}{1}\cdot\frac{4}{3}\cdot\frac{6}{5}\cdot\ \cdots\ \cdot\frac{22}{21}$

(أ) نضع أولا $SUM = 0$ ثم نجمع بالتتابع $1/K$ إلى SUM لكل من $K = 1, 3, 5, \ldots, 21$. يعطى الشكل ٤ — ٢٤ (أ) خريطة سير العمليات لاحظ أننا نضع $K = 1$ أولا ثم نزيد K بمقدار 1 فى كل مرة لأن المجموع يشمل الأعداد الفردية فقط . وعلاوة على ذلك ، نختبر إذا كانت $K \leq 21$ حيث أننا لانريد أن نجمع أى شىء بعد جمع 1/21 . عند ترجمة خريطة سير العمليات إلى الفورتران ، يجب أن نكتب .

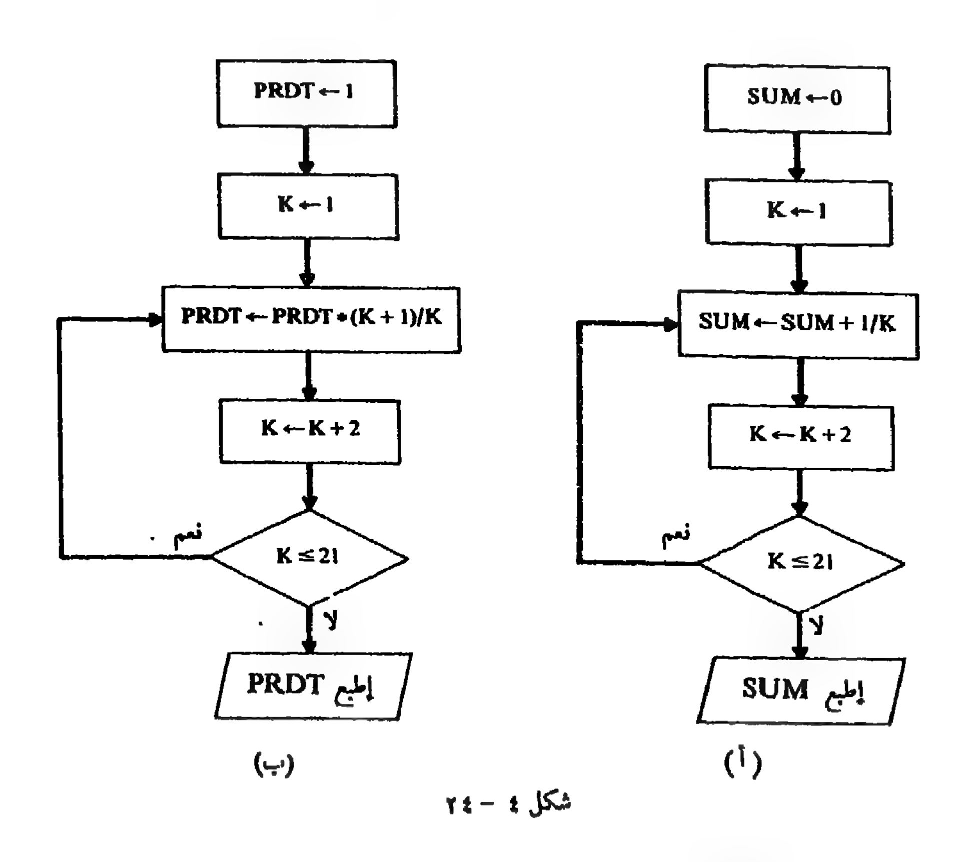
```
1.0/FLOAT(K)
```

بدلا من كتابه $1/K$ حيث أننا لانريد أن نجرى قسمة صحيحة . يظهر البرنامج كما يلى :

```
      SUM = 0.0
      K = 1
11    SUM = SUM + 1.0/FLOAT(K)
      K = K + 2
      IF(K.LE.21) GO TO 11
      WRITE(6, 20) SUM
20    FORMAT(1X, 'THE SUM IS', 2X, F8.5)
      STOP
      END
```

(ب) نجعل PRDT = 1 أولا بعد ذلك نضرب PRDT في (K + 1)/K لكل من K = 1, 3,....,21 . يبين شكل ٤ — ٢٤ (ب) خريطة سير العمليات المطلوبه لاحظ التشابه مع الجزء (أ) . وفيما يلي البرنامج :

```
      PRDT = 1.0
      K = 1
13    PRDT = PRDT*FLOAT(K + 1)/FLOAT(K)
      K = K + 2
      IF(K.LE.21) GO TO 13
      WRITE(6, 30) PRDT
30    FORMAT(1X, 'THE PRODUCT IS', 2X, F8.5)
      STOP
      END
```



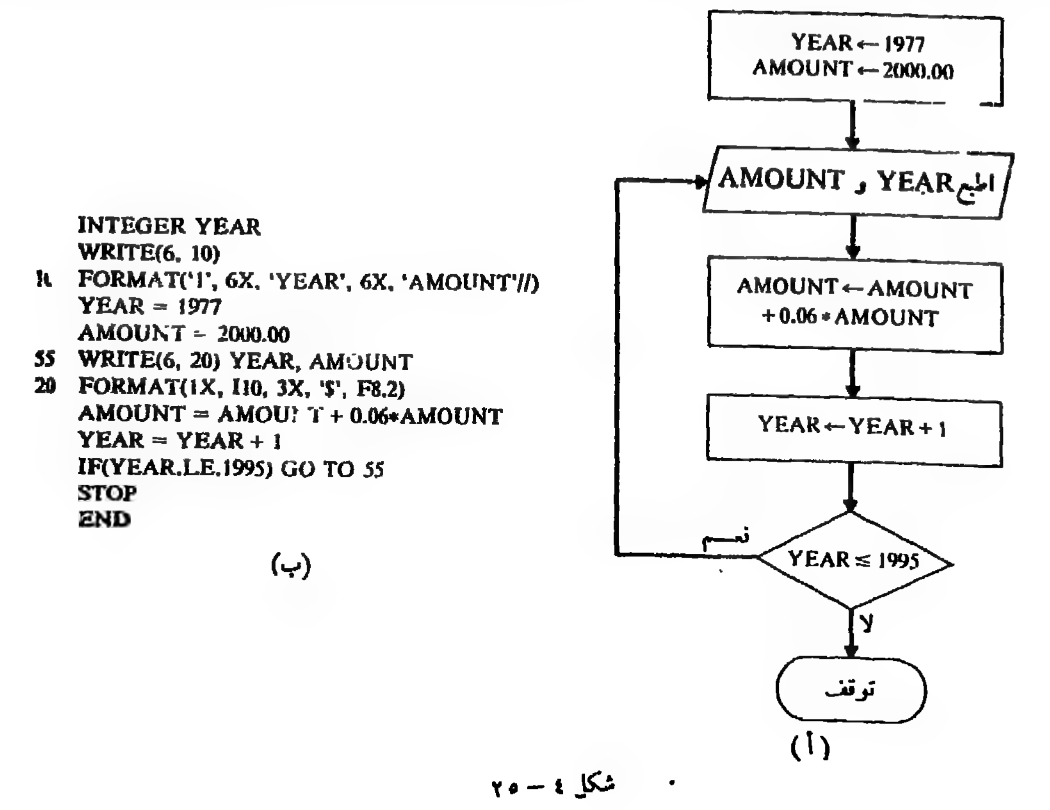
شكل ٤ — ٢٤

٤ — ١١ إفرض أنه تم إيداع مبلغ $2000.00 في حساب توفير في سنة 1977 وافرض أن البنك يدفع 6 في المائة فائدة مركبة سنوياً على الحساب . اكتب برنامجاً يطبع (السنة) YEAR و (القيمة) AMOUNT لهذا الحساب إلى سنة 1995 .

تزاد AMOUNT كل سنة بمقدار 6 في المائة وبذلك يكون التخصيص

$$AMOUNT \leftarrow AMOUNT + 0.06 * AMOUNT$$

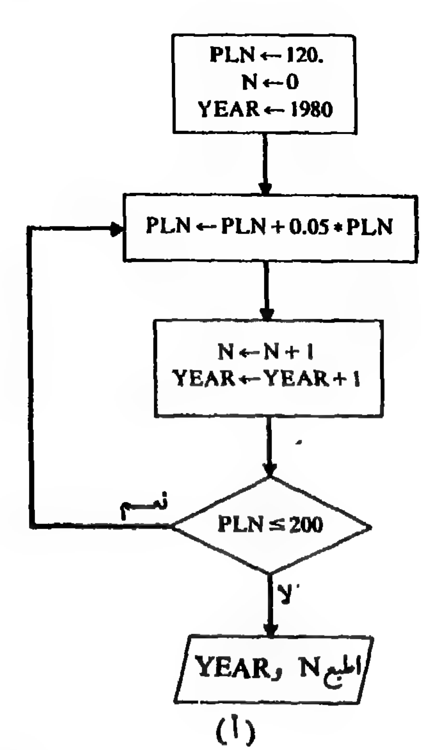
نكرر هذه العملية طالما YEAR ≤ 1995 يبين شكل ٤ — ٢٥ خريطة سير العمليات للبرنامج ومايقابلها بالفورتران .



شكل ٤ — ٢٥

```
      INTEGER YEAR
      PLN = 120.0
      N = 0
      YEAR = 1980
100   PLN = PLN + 0.05*PLN
      N = N + 1
      YEAR = YEAR + 1
      IF(PLN.LE.200.) GO TO 100
      WRITE(6, 10) N, YEAR
 10   FORMAT(1X, I3, 3X, I4)
      STOP
      END
```

(ب)



شكل ٤ — ٢٦

٤ – ١٢ إفرض أن تعداد بلد PLN فى سنة 1980 يبلغ 120 مليون نسمة وافرض أن التعداد يزداد بنسبة 5 فى المائة كل سنة . اكتب البرنامج الذى يحدد عدد السنوات N والسنة YEAR عندما يتعدى التعداد 200 مليون نسمة .

يزداد PLN كل سنة بنسبة 5 فى المائة . وبذلك فإن التخصيص يتكرر طالما PLN ≤ 200 .

$$PLN \leftarrow PLN + 0.05*PLN \qquad (١)$$

تظهر خريطة سير العمليات للبرنامج فى شكل ٤ – ٢٦ (أ). لاحظ أننا نطبع N و YEAR فقط بعد أن يتعدى PLN الرقم 200. تظهر ترجمة البرنامج بالفورتران فى شكل ٤ – ٢٦ (ب) .

(يمكن أن تكتب PLN*1.05 بدلا من PLN + 0.05*PLN)

٤ – ١٣ افرض أن كل بطاقة من مجموعة البطاقات تحتوى على عدد حقيق . أضيفت بطاقة مقدمة تحتوى على عدد البطاقات فى المجموعة N نريد أن نجد أكبر عدد للبطاقات فى المجموعة .

(أ) ما هى التغيرات التى يجب عملها فى الشكل ٤ – ١٧ لحل المسألة . (يبين شكل ٤ – ١٧ مجموعة بطاقات عددها 100 بطاقة فقط) .

(ب) اكتب البرنامج .

(أ) نحتاج إلى إضافة READ N فقط فى بداية خريطة سير العمليات ، وتغيير I ≤ 99 إلى I ≤ N — 1 فيما عدا ذلك ستكون خريطة سير العمليات مطابقة للخريطة السابقة .

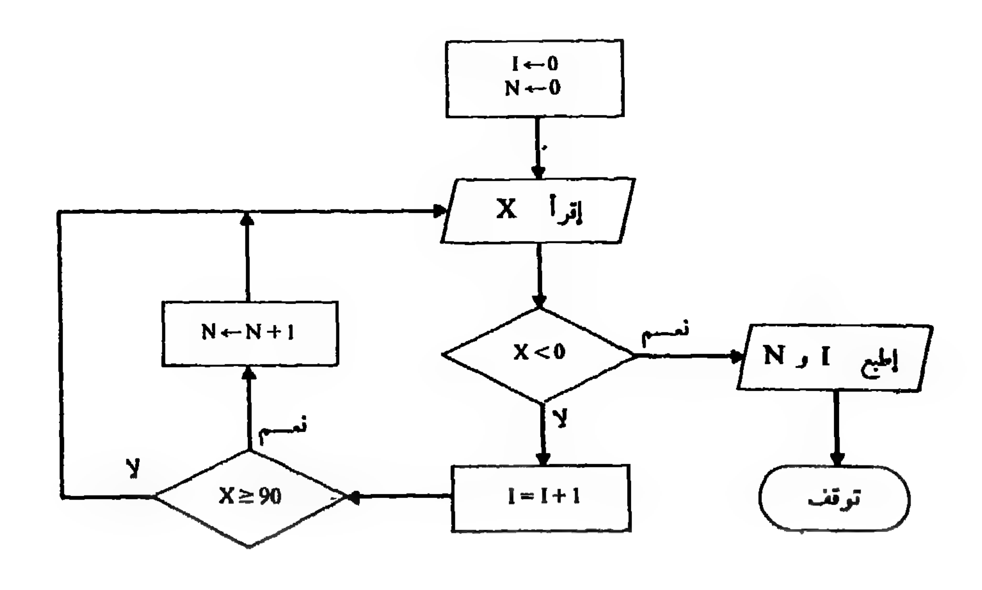
(ب) يظهر البرنامج فيما يلى . لاحظ أننا احتجنا لجملة نوع لتعلن أن LAR متغير حقيق (REAL) .

```
C
C      PROGRAM FINDING LARGEST NUMBER
C
       REAL LAR
       READ(5, 10) N
 10    FORMAT(I6)
       READ(5, 20) LAR
       I = 1
100    READ(5, 20) X
 20    FORMAT(F12.2)
       IF(LAR.GE.X) GO TO 200
       LAR = X
200    I = I + 1
       IF(I.LE.N - 1) GO TO 100
       WRITE(6, 30) LAR
 30    FORMAT(1X, 'THE LARGEST NUMBER IS', 2X, F12.2)
       STOP
       END
```

٤ – ١٤ افرض أن كل بطاقة من مجموعة بطاقات تحتوى على درجة اختبار لطالب . وكانت المجموعة لها بطاقة خلفية مثقب عليها عدد سالب . ارسم خريطة سير العمليات واكتب برنامج الفورتران الذى يحسب عدد الطلبة I الذين أدوا الاختبار ويحسب أيضاً العدد N وهى عدد درجات الاختبار التى تساوى أو تزيد عن 90 .

شكل ٤ — ٢٧ يبين خريطة سير العمليات . لاحظ أن I أعطيت قيمة إبتدائية صفر . وفيما يلي مايقابل الخريطة بالفورتران.

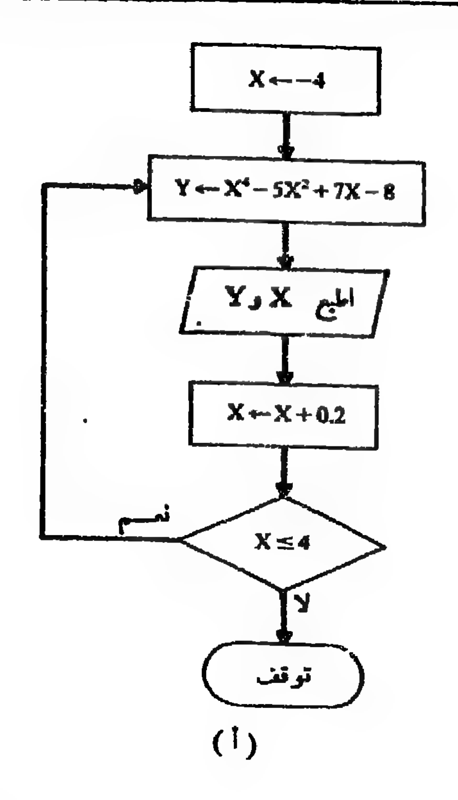
```
C
C        PROGRAM SCORES
C
         I  = 0
         N = 0
  100    READ(5, 10) X
   10    FORMAT(F10.1)
         IF(X.LT.0.0) GO TO 200
         I = I + 1
         IF(X.GE.90.0) N = N + 1
         GO TO 100
  200    WRITE(6, 20) I, N
   20    FORMAT('0', I10, 2X, 'STUDENTS TOOK THE TEST'/
     1      '0', I10, 2X 'SCORED ABOVE 90')
         STOP
         END
```



شكل ٤ — ٢٧

٤ — ١٥ إدرس المعادلة $y = x^4 - 5x^2 + 7x - 8$. ارسم خريطة سير العمليات واكتب برنامج الفورتران الذى يحسب قيمة y لقيم x إبتداء من 4 — إلى 4 بزيادة قدرها 0.2 كل مرة . اطبع قيمة x وقيمة y المناظرة على أسطر مختلفة .

أولا نخصص القيمة 4 — إلى X وبعد حساب Y وطباعة Y و X نزيد قيمة X بمقدار 0.2 . ونكرر العملية طالما $X \leq 4$. تظهر خريطة سير العمليات للنظام الحسابي (الخوارزم) فى شكل ٤ — ٢٨ (أ) ويظهر مايقابله بالفورتران فى شكل ٤ — ٢٨ (ب) .



(أ)

```
      X = −4.0
100   Y = X**4 − 5.0*X**2 + 7.0*X − 8.0
      WRITE(6, 10) X, Y
10    FORMAT(6X, 'X =', F5.1, 3X, 'Y =', 2X, F6.2)
      X = X + 0.2
      IF(X.LE.4.0) GO TO 100
      STOP
      END
```

(ب)

شكل ٤ – ٢٨

مسـائل تكميلية

جمل IF

٤ – ١٦ اكتب الجمل التالية بالفورتران

(أ) توقف إذا كانت $A > B$

(ب) إذهب إلى الجملة رقم 20 إذا كانت $J \neq K + 3$

(ﺟ) إذهب إلى الجملة رقم 30 إذا كانت $A + B^2 < 100$

(د) إذهب إلى الجملة رقم 40 إذا كانت $X - Y \geq 50$

(ﻫ) إذهب إلى الجملة رقم 50 إذا كانت $I \not\leq 4$

(و) توقف إذا كانت $J \leq K$

٤ – ١٧ اكتشف الأخطاء ، إن وجدت في كل من الجمل الآتية :

(أ) IF(A = B) GO TO 50

(ب) IF(X GT Y) STOP

(ﺟ) IF(B**2 − A*C) STOP

(د) IF(X.LT.Y + Z) 10, 15, 20

(ﻫ) IF(X.LE.100) GO TO K

(و) IF(A − 100) 10, 20, 30,

(ز) IF(X.GE.Y) GO TO 55

(ح) IF(INTEREST.LT.AMOUNT STOP

٤ – ١٨ نفذ كلا مما يأتى بطريقتين ، مرة باستخدام جمل IF المنطقية ومرة أخرى باستخدام جمل IF الحسابية .

(أ) انقل التحكم إلى الجملة رقم 100 إذا كانت $A + B^2 > 100$ وانقل التحكم إلى الجملة رقم 200 فيما عدا ذلك .

(ب) ضع $K = 1$ إذا كانت $A - B \leq 100$ وضع $K = 2$ فيما عدا ذلك .

(جـ) انقل التحكم إلى الجملة 10 أو 20 أو 30 تبعاً للشروط $J < K$ و $J = K$ و $J > K$.

٤ – ١٩ أعد كتابة الآتى : باستعمال جمل IF الحسابية . (هنا S و T جمل قابلة التنفيذ)

(أ)
```
IF(X.GT.Y) S
T
```

(ب)
```
IF(I.EQ.J) S
T
```

(جـ)
```
IF(X.LE.Y) S
T
```

(د)
```
IF(A.GE.B) S
T
```

٤ – ٢٠ افرض أن J و K تحتويا 3 و 5 على الترتيب . أوجد قيمة J النهائية بعد كل جزء من البرنامج .

(أ)
```
   IF(J.GE.K) J = J + 2
   J = J + 2
```

(ب)
```
   IF(5*J.EQ.3*K) J = J + 2
   J = J + 2
```

(جـ)
```
   IF(J - K) 10, 10, 20
10 J = K
20 J = J + 2
```

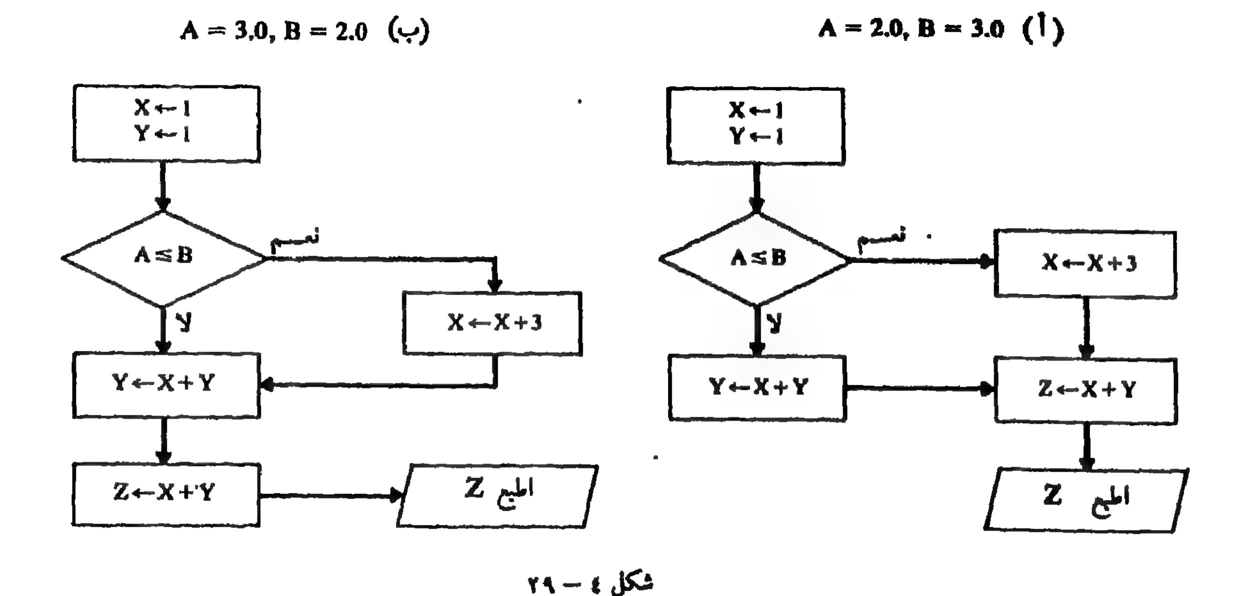
(د)
```
   IF(J.LT.K - 1) GO TO 10
   J = J + 2
10 J = J + K
```

(هـ)
```
   IF(J.GE.K + 1) GO TO 10
   J = J + 2
10 J = J + K
```

(و)
```
   IF(2*J - K) 10, 10, 20
10 J = K
20 J = J + 2
```

٤ – ٢١ ترجم خريطتى سير العمليات فى شكل ٤ – ٢٩ إلى جزء من برنامج فورتران . وأوجد قيمة Z فى كل حالة إذا كانت .

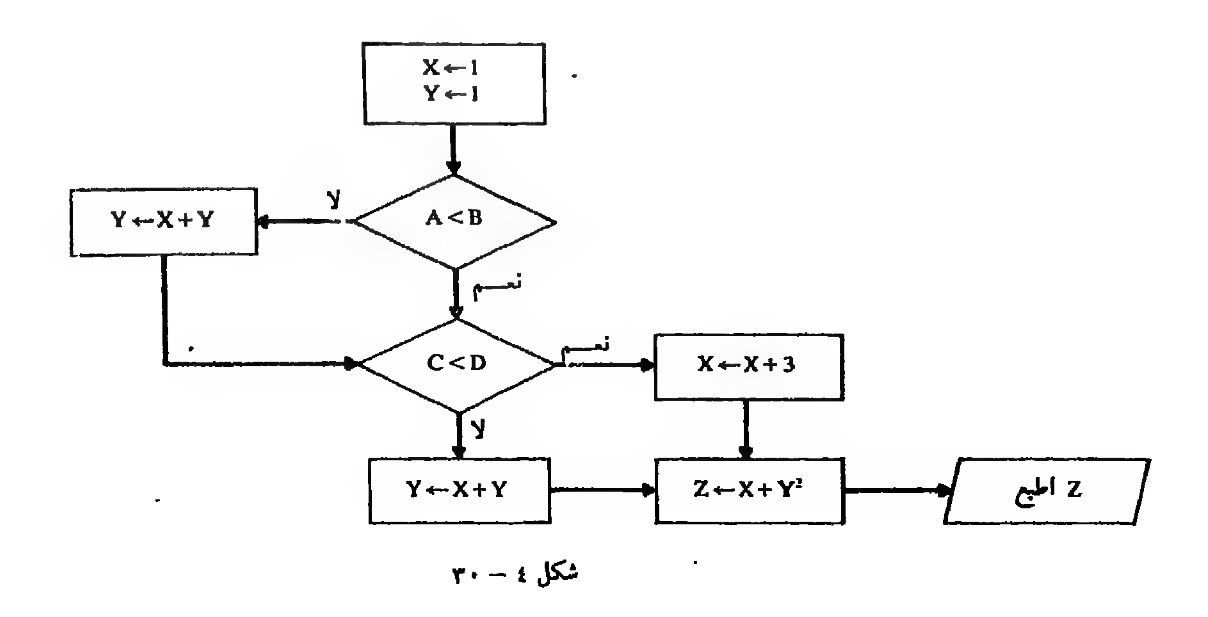
(أ) A = 2.0, B = 3.0 (ب) A = 3.0, B = 2.0



شكل ٤ – ٢٩

٤ — ٢٢ ترجم خريطة سير العمليات فى شكل ٤ — ٣٠ إلى جزء من برنامج فورتران ثم أوجد قيمة Z إذا كانت :

(أ) A = 2.0, B = 3.0, C = 3.0, D = 2.0
(ب) A = 3.0, B = 2.0, C = 3.0, D = 2.0
(ح) A = 3.0, B = 2.0, C = 2.0, D = 3.0
(د) A = 2.0, B = 3.0, C = 2.0, D = 3.0



شكل ٤ — ٣٠

جمل GO TO المحسوبة

٤ — ٢٣ اكتشف الأخطاء ، إن وجدت فى كل جملة من جمل GO TO المحسوبة التالية :

(أ) GO TO (35, 17, 17, 46) JIM
(ب) GO TO (42, 0, 88, 1234), LAST
(ح) GO TO (34, 34, 58, 58, 34), N237K
(د) GO TO (234, 2345678, 7654), J
(ه) GO TO (34, 82, 56, 77), TOM

٤ — ٢٤ اكتب مجموعة من جمل IF المنطقية المكافئة لـ :

GO TO (47, 33, 55, 77), K

٤ — ٢٥ اكتب جملة IF حسابية مكافئة لـ :

GO TO (20, 30, 40), JIM

٤ – ٢٦ أوجد رقم الجملة التى ينتقل إليها التحكم بعد كل جزء من برامج الفورتران التالية :

(أ)
```
J = 3
GO TO (21, 31, 41, 51), J
```

(جـ)
```
INTEGER TYPE
K = 3
TYPE = 2 + K
GO TO (21, 31, 41, 51), TYPE
```

(ب)
```
I = 2
NEXT = 4 - I
GO TO (21, 31, 41, 51), NEXT
```

(د)
```
INTEGER TYPE
K = 2
TYPE = 2*K
GO TO (21, 31, 41, 51), TYPE
```

بـرامج :

٤ – ٢٧ اكتب البرنامج الذى يطبع الأعداد الصحيحة الموجبة من 1 إلى 300 بحيث تظهر ثلاثة أعداد فى كل سطر ، أى ، لكى يبدو الخرج كما يلى :

```
  1     2     3
  4     5     6
..................
298   299   300
```

٤ – ٢٨ اكتب البرنامج الذى يحسب ، إلى مكانين عشريين ، المجموع التالى

$$1/2 + 2/3 + 3/4 + \cdots + 99/100$$

٤ – ٢٩ اكتب برنامج الذى يقرأ عددا صحيحاً موجباً N حيث $N \geq 10$ ثم يحسب المجاميع التالية مقربة إلى مكانين عشريين .

(أ) $1 + 1/2 + 1/3 + \cdots + 1/N$

(ب) $1 - 1/2 + 1/3 - \cdots \pm 1/N$

بفرض أن A سبق تعريفها . احسب أيضا ما يلى

(جـ) $\frac{1}{1+A} + \frac{1}{1+2A} + \frac{1}{1+3A} + \cdots + \frac{1}{1+N \cdot A}$

٤ – ٣٠ اكتب برنامجاً يقرأ عدداً صحيحاً موجباً N حيث $N \geq 10$ ثم يحسب حاصل الضرب التالى مقرباً إلى خمسة أرقام عشرية :

$$\frac{1}{1^2} \cdot \frac{3}{2^2} \cdot \frac{5}{3^2} \cdot \cdots \cdot \frac{2N-1}{N^2}$$

٤ – ٣١ اكتب برنامج فورتران لحساب إجمالى الدخل GROSS إذا كان كل من ID (رقم الموظف) ، RATE (الأجر فى الساعة) و HOUR (عدد الساعات التى عملها الموظف فى أسبوع) مخزنة فى الذاكرة ، مع ملاحظة أن حيث الأجر الإضافى يدفع على أساس مرة ونصفاً من الأجر .

(الأجر الإضـــافى هو مازاد عن ٤٠ ساعة)

٤ – ٣٢ تحسب عمولة المندوب على المبيعات الإجمالية SALES كالآتى :

(أ) لا تكون هناك عمولة ، إذا كانت SALES < $50

(ب) العمولة = ١٥ ٪ من المبيعات SALES إذا كانت $ 50 ≤ SALES ≤ $500

(ﺣ) العمولة = 50 $ + 8 % من المبيعات إذا كانت المبيعات تزيد على $500 . اكتب البرنامج الذى يقرأ المبيعات الكلية ومنها يحسب العمولة .

٤ – ٣٣ افرض أننا أودعنا AMOUNT يساوى $1,000.00 فى حساب توفير بنسبة ربح مركبة مقدارها 7 فى المائة سنوياً .

(أ) اكتب البرنامج الذى يطبع AMOUNT لحساب كل سنة ولمدة 20 سنة .

(ب) اكتب البرنامج الذى يحدد السنوات التى يستغرقها الحساب حتى تتعدى AMOUNT المقدار $5,000.00 .

٤ – ٣٤ افرض أننا قرأنا AMOUNT لوديعة ، بفائدة (مركبة سنوياً) RATE وعدد سنين N . اكتب البرنامج الذى يطبع قيمة الحساب VALUE كل سنة لعدد N من السنوات (إختبر البرنامج مع البيانات فى المسألة ٤ – ٣٣) .

٤ – ٣٥ تعرف سلسلة فيبوناس كما يلى :

1, 1, 2, 3, 5, 8, 13, 21, . . .

ومنها نجد أن الحدين الأولين لها القيمة 1 وكل حد بعد ذلك هو مجموع الحدين السابقين :

$1 + 1 = 2, 1 + 2 = 3, 2 + 3 = 5, 3 + 5 = 8, \ldots$

(أ) اكتب البرنامج الذى يطبع سلسلة فيبوناس دون أن يتجاوز 10,000

(ب) اكتب البرنامج الذى يطبع أول 50 حد من سلسلة فيبوناس .

٤ – ٣٦ اجعل J و K أعدادا صحيحة موجبة بحيث تكون J ≤ K . وتعرف صورة سلسلة فيبوناس العامة بإعتبار J الحد الأول ، K الحد الثانى ، وكل حد بعد ذلك هو مجموع الحدين السابقين . اكتب البرنامج الذى يقرأ J و K ويطبع أول 50 حد من سلسلة فيبوناس العامة .

٤ – ٣٧ أفرض أن كل بطاقة فى المجموعة تحتوى على عدد حقيقى موجب (قيمته أقل من 1000) والمجموعة لها بطاقة خلفية .

(أ) اكتب البرنامج الذى يجد العدد الأكبر فى المجموعة (قارن مع المسألة ٤ – ٣) ، والتى تستعمل بطاقة مقدمة .

(ب) اكتب البرنامج الذى يجد أصغر رقم فى المجموعة .

(ﺣ) اكتب البرنامج الذى يجد الرقم الذى يلى أكبر رقم فى المجموعة (اعتبر أن المجموعة بها أكثر من رقم واحد) .

٤ – ٣٨ ترجم شكل ٤ – ١٩ (ب) إلى الفورتران . والشكل يحسب المتوسط الحسابى لمائة عدد .

٤ – ٣٩ ترجم شكل ٤ – ٢١ (أ) إلى الفورتران ، والشكل يحسب المتوسط الحسابى لمجموعة من الأعداد غير السالبة . المجموعة لها بطاقة خلفية مثقب بها عدد سالب .

٤ ـ ٤٠ افرض أن كل بطاقة فى مجموعة تحتوى على الدرجة الصحيحة لطالب فى اختبار . (أضيفت بطاقة خلفية تحتوى على عدد سالب) خصص لدرجة الاختبار T المتغيرات A أو B أو C أو D و F تبعاً للآتى :

$T \geq 90$ أو $80 \leq T < 90$ أو $70 \leq T < 80$ أو $60 \leq T < 70$ أو $T < 60$ على الترتيب ، اكتب البرنامج الذى يوجد :

(أ) عدد التلاميذ الذين أدوا الاختبار (I) .

(ب) عدد كل من A و B و C و D و F .

(حـ) عدد الأوراق الدرجات النهائية ، أى العدد K من الأوراق الحاصلة على 100 درجة . (المسألة العامة ٤ ـ ١٤) .

٤ ـ ٤١ افرض ، فى الوقت الحالى ، أن تعداد بلدين A و B هو 52 و 85 مليون نسمة على الترتيب ، وافرض أن معدلات النمو السكانى فى A و B هى 6 فى المائة و 4 فى المائة على الترتيب . اكتب البرنامج الذى يطبع التعداد (إلى أقرب ألف) للبلدين A و B كل سنة إلى أن يتجاوز تعداد A تعداد B . ثم أوجد عدد السنوات N التى استغرقها تعداد A حتى يتجاوز تعداد B

٤ ـ ٤٢ تماماً مثل مسألة ٤ ـ ٤١ ، ولكن إذا أعطيت بيانات التعداد ومعدلات النمو للبلدين A و B وليكن POPA و POPB و RATEA و RATEB على الترتيب ، على بطاقات بيانات . (اختبر البرنامج ببيانات مسألة ٤ ـ ٤١) .

٤ ـ ٤٣ (أ) اكتب البرنامج الذى يطبع كل الأعداد الصحيحة الموجبة أقل من 100 مع حذف تلك الأعداد الصحيحة التى تقبل القسمة على 7 . أى :

1, 3, 5, 9, 11, 15, 17, 19, 23, . . . , 97, 99

(ب) أعد كتابة البرنامج السابق حتى يكون هناك أربعة أرقام على كل سطر ، أى لكى يبدو الخرج كما يلى :

1	3	5	9
11	13	17	19
23	25	27	29

. . . .

(٢)

٤ ـ ٤٤ اكتب البرنامج الذى يقرأ العدد الصحيح الموجب N حيث $10 \leq N \leq 1000$ ثم يطبع ، على صفحة جديدة ، العدد N وكل الأعداد التى تقبل القسمة عليها حتى يظهر الخرج كما يلى (إذا كانت $N = 12$ مثلا)

```
DIVISORS of 12
     1
     2
     3
     4
     6
     12
```

٤ ـ ٤٥ تجهز بطاقة لكل طالب فى فصل للدراسات التكميلية توضح عليها سن الطالب والجنس وموقفه من الفصل والحالة الاجتماعية ، وتبين فى أكواد كالتالى :

(١) 1 للأنثى و 2 للذكر

(٢) 1 للطالب المستجد ، 2 بالسنة الثانية ، 3 بالسنة الثالثة 4 بالسنة الرابعة ، 5 للخريج ، 6 لغير المسجل بدراسة نظامية .

(٣) 1 للأعزب ، 2 للمتزوج .

تستعمل بطاقة خلفية لإنهاء المجموعة . أكتب برنامجاً لحساب النسب المئوية لكل ما يأتى : (أ) عدد الذكور (ب) عدد الإناث . (ﺣ) عدد الطلبة الخريجين (د) عدد الطلبة الذين تتجاوز أعمارهم 30 عاماً .

٤ — ٤٦ اكتب برنامجاً لحساب النسب المئوية مستخدماً البيانات المعطاة فى المسألة ٤ — ٤٥ السابقة لما يأتى : (أ) عدد الخريجين المتزوجين (ب) عدد إناث الفصول العليا (السنوات الثالثة والرابعة) (ﺣ) عدد غير المتخرجين فوق 30 سنة .

٤ — ٤٧ كل بطاقة فى مجموعة تحتوى على ثلاثة أعداد موجبة A ، B ، C والمجموعة لها بطاقة خلفية .

(أ) اكتب برنامج لتحديد ما إذا كانت A ، B ، C تشكل أضلاع مثلث. إذا كانت الإجابة نعم ، احسب محيط المثلث ، أما إذا كانت لا ، فاطبع الرسالة "NOT A TRIANGLE" (تلميح : A و B و C يمكن أن تشكل مثلثاً إذا كان كل ضلع أقل من مجموع الضلعين الآخرين أى إذا كانت $A < B + C$ ، $B < A + C$ ، $C < A + B$)

(ب) اكتب برنامجاً لتحديد ما إذا كانت A ، B ، C تشكل أضلاعاً : (١) لمثلث متساوى الأضلاع (ثلاثة أضلاع متساوية) (2) لمثلث متساوى الساقين (ضلعان متساويان) (3) لمثلث قائم الزاوية (الوتر٢ = ضلع٢ + ضلع٢)

٤ — ٤٨ يعطى مثال ٤ — ٥ البرنامج الذى يحدد ما إذا كان العدد الصحيح $N > 2$ عدداً أولياً ويتم ذلك باختبار إذا كانت N تقبل القسمة على أى من الأعداد الصحيحة 2 ، 3 ، ... ... إلى N/2 . فى الواقع إذا كانت N ليست عدداً أولياً ، فإنها تقبل القسمة على قاسم $\leq \sqrt{N}$.

(أ) عدل البرنامج بحيث يختبر ما إذا كانت N تقبل القسمة على أى من الأعداد الصحيحة 2 ، 3 ، . . . إلى $\sqrt{2}$ فقط

(ب) عدل البرنامج بحيث يختبر ما إذا كانت N تقبل القسمة على 2 أو على أى عدد صحيح مفرد $\leq \sqrt{N}$ فقط

٤ — ٤٩ ترجم الجملة الآتية إلى الفورتران .

```
IF  1 ≤ X ≤ 2  THEN  K = 1
               ELSE  K = 2
```

(تلميح : الشرط $1 \leq X \leq 2$ يعنى $1 \leq X$ ، $X \leq 2$ ويمكن أن تنفذ بواسطة جمل IF المنطقية) .

اجابات للمسائل التكميلية المختارة

٤ — ١٦

(أ)	IF(A.GT.B) STOP	(د)	IF(X – Y.GE.50.0) GO TO 40
(ب)	IF(J.EQ.K + 3) GO TO 20	(ﻫ)	IF(I.NE.4) GO TO 50
(ﺣ)	IF((A + B**2).LT.100.0) GO TO 30	(و)	IF(J.LE.K) STOP

٤ – ١٧ (أ) .EQ. بدلا من =

(ب) .GT. بدلا من .GT .

(ج) $B^2 - AC$ ليست تعبيراً مترابطاً .

(د) X.LT.Y + Z ليست تعبيراً حسابياً .

(هـ) 1000 بدلا من 100 إذا كانت X حقيقية.

(و) لا توجد أخطاء ، رغم أننا نفضل A — 100.0 .

(ز) لايوجد أخطاء .

(ح) يجب أن تكون هناك أقواس بين AMOUNT و STOP ؛

INTREST لها أكثر من ستة حروف ؛ INTEREST و AMOUNT لها أنواع مختلفة .

٤ – ١٨ (أ)

```
      IF(A + B**2 - 100.0) 200, 200, 100
```
or
```
      IF(A + B**2.GT.100.0) TO TO 100
200   ..................................
```

(ب)

```
      IF(A - B - 100.0) 10, 10, 50
50    K = 2
      GO TO 20
10    K = 1
20    ................. . . ..
```
or
```
      IF(A - B.LE.100.0) GO TO 10
      K = 2
      GO TO 20
10    K = 1
20    ..........................
```

(ج)

```
      IF(J - K) 30, 20, 10
```
or
```
      IF(J.LT.K) GO TO 10
      IF(J.EQ.K) GO TO 20
      IF(J GT.K) GO TO 30
```

٤ – ١٩ (أ)

```
      IF(X - Y) 20, 20, 10
10    S
20    T
```

(ب)

```
      IF(I - J) 20, 10, 20
10    S
20    T
```

(ج)

```
      IF(X - Y) 10, 10, 20
10    S
20    T
```

(د)

```
      IF(A - B) 20, 10, 10
10    S
20    T
```

٤ – ٢٠

(أ) 5 (ب) 7 (ج) 7 (د) 8 (هـ) 10 (و) 5

٤ – ٢١

(أ) 9.0 و 3.0 (ب) 5.0 و 3.0

٤ - ٢٢

(أ) 5.0 (ب) 10.0 (جـ) 8.0 (د) 5.0

٤ - ٢٣ (أ) لاتوجد فصلة قبل JIM .

(ب) غير مسموح بـ 0 .

(جـ) لاتوجد أخطاء .

(د) 2345678 كبيرة أكثر من اللازم لرقم جملة .

(هـ) TOM يجب أن تكون متغيراً صحيحاً .

٤ - ٢٤

```
IF(K.EQ.1) GO TO 47
IF(K.EQ.2) GO TO 33
IF(K.EQ.3) GO TO 55
GO TO 77
```

٤ - ٢٥

```
IF(JIM - 2) 20, 30, 40
```

٤ - ٢٦

(أ) 41

(ب) 31

(جـ) 51

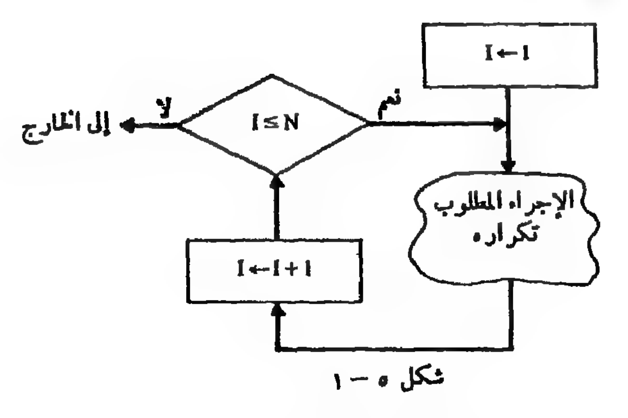
(د) خطأ حيث أن TYPE تحتوى على 5 .

الفصل الخامس

حلقــات DO التكرارية

٥ — ١ مقدمة

يمكننا نقل التحكم الذى تمت مناقشته فى الفصل الرابع من تنفيذ مجموعة من التعليمات عدة مرات . نذكر أننا استعملنا عداد I بصحبة جملة IF لخروج من الحلقات التى تمت مناقشتها فى القسمين ٤ — ٥ و ٤ — ٧ . توضح خريطة سير العمليات فى شكل ٥ — ١ . ميكانيكية التحكم لتكرار أجراء عدد N من المرات . وهى أن العداد I قد أعطى القيمة 1 فى البداية . وبعد كل تكرار . تزاد قيمة هذا العداد بمقدار 1 ثم قبل تكرار الحلقة مرة ثانية يسأل هل I ≤ N? تنتهى الدورة بمجرد تجاوز I القيمة N.



شكل ٥ — ١

وحيث أن تكرار إجراء شيء عملية جوهرية عند كتابة برامج الحاسب ، فن المفيد أن يكون لدينا أمر مطول (macrolike) مثل :

DO....WHILE 1 ≤ I ≤ N

DO

م DO [⋮ الإجراء المطلوب تكراره ⋮]

طالما 1 ≤ I ≤ N

أو أمر مطول مثل DO FOR 1 ≤ I ≤ N...REPEAT.

DO FOR 1 ≤ I ≤ N

جـم DO [⋮ الإجراء المطلوب تكراره ⋮]

REPEAT

يحتوى الفورتران غير الهيكلى على مثل هذا النوع من الأوامر ألا وهى جملة DO . وحيث أن جملة DO هى من أقوى تركيبات الفورتران فقد كرسنا هذا الفصل بالكامل لمناقشة هذا الأمر المطول . (ونعنى بالأمر المطول . أمراً مفرداً يستحضر قائمة من الأوامر التى سبق تحديدها) .

يحتوى الفورتران الهيكلى ، الذى يناقش فى الفصل الثانى عشر ، على أنواع أخرى من أوامر DO .

٥ — ٢ جمل CONTINUE

نقدم أولا جملة جديدة قابلة التنفيذ وهى جملة CONTINUE والتى تتكون ببساطة من الكلمة

CONTINUE

ستتضح أسباب وجود هذه الجملة بعد ذلك فى قسم ٥ — ٨ ، ولكن وظيفتها (عملها) تتصف تماماً باسمها — وهى أكل التنفيذ . تسمى أيضاً جملة CONTINUE جملة قابلة التنفيذ زائفة حيث لا يتولد عنها أمر بلغة الآلة .

تسمح معظم المترجمات باستعمال جملة CONTINUE فى أى مكان من البرنامج ، بينما يتطلب البعض أن تحمل جمل CONTINUE رقم .

٥ — ٣ استخدامات بسيطة لجملة DO

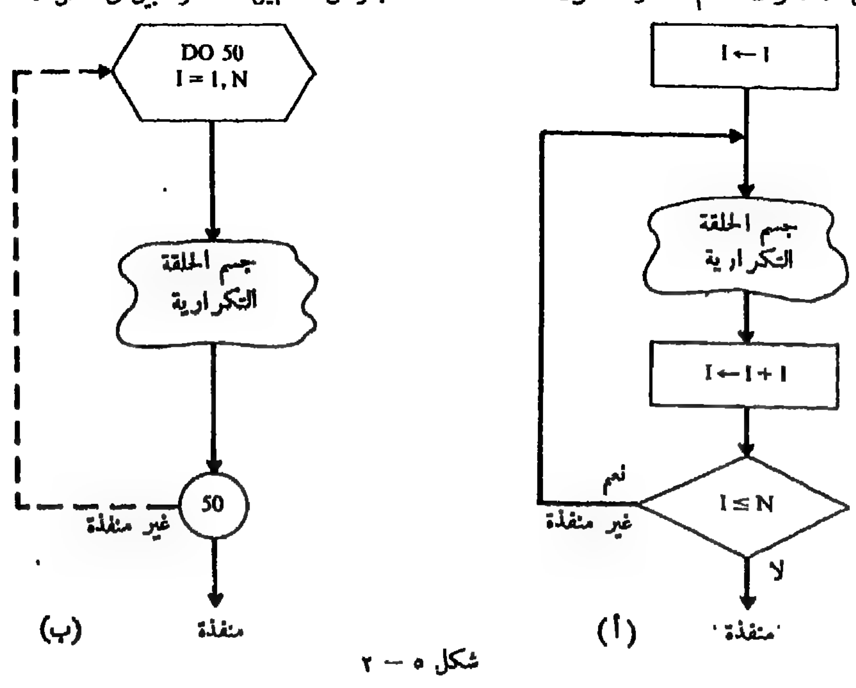
لتكرار إجراء ما عدد N من المرات ، يمكن أن نستخدم الشكل التالى الشائع الاستخدام لجملة DO .

```
      DO 50 I = 1, N
          [     :           ]
          [BODY OF DO LOOP  ]
          [     :           ]
   50 CONTINUE
```

يأمر زوج الجمل السابق (DO-CONTINUE) الحاسب أن يكرر تنفيذ الجمل الواقعة بين DO وCONTINUE (وتسمى الحلقة التكرارية) . يخصص للمتغير الصحيح I (ويسمى المتغير الدليل) فى البداية القيمة 1 وبعد كل تكرار ، تزاد قيمة I بمقدار 1 . يستمر التكرار طالما $I \leq N$ وعندما تصبح $I > N$ تنفذ الجملة التى تلى جملة CONTINUE بعد ذلك . (نلاحظ أن رقم جملة CONTINUE تم اختياره عشوائياً)

ملاحظة : ولسهولة القراءة ، نحرك عادة جسم الحلقة التكرارية للداخل ولا يؤثر هذا على البرنامج حيث أن المسافات الخالية فى الفورتران يتغاضى عنها الحاسب .

توضح خريطة سير العمليات فى شكل ٥ — ٢ (أ) المعنى الدقيق لجملة DO السابقة (العليا) . وحيث أن جملة DO تتضمن عدة صناديق ، فسوف نقدم صندوقاً مطولا مكافئاً ، ذلك بغرض التسهيل كما هو مبين فى شكل ٥ — ٢ (ب)

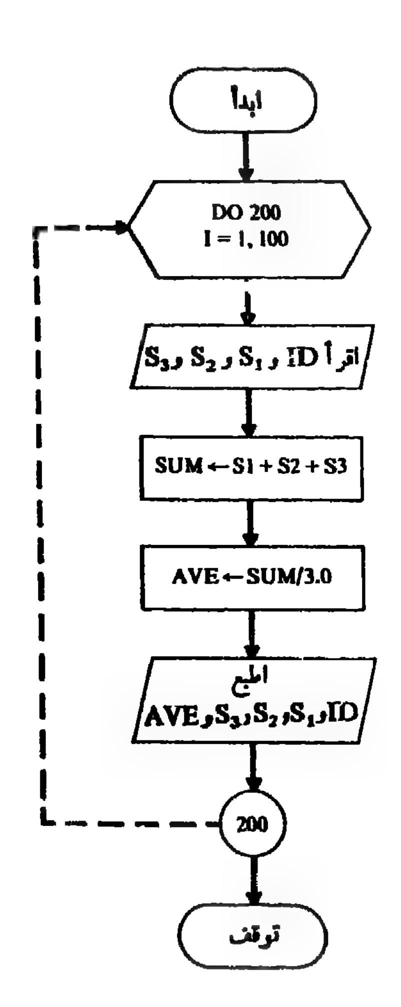


شكل ٥ — ٢

مثال ٥ — ١

تذكر مثال ٤ — ٣ ، الذي يحسب متوسط ثلاث درجات اختبار لكل طالب فى فصل به 100 طالب (تثقب درجات اختبار الطالب الثلاث ورقمه ID على بطاقة بيانات واحدة) .

ونعيد كتابة البرنامج باستعمال جملة DO ويبين شكل ٥ — ٣ خريطة سير العمليات وما يقابلها فى الفورتران . لاحظ كيف يبدو جسم الحلقة التكرارية واضحاً بسبب تحريكه للداخل .



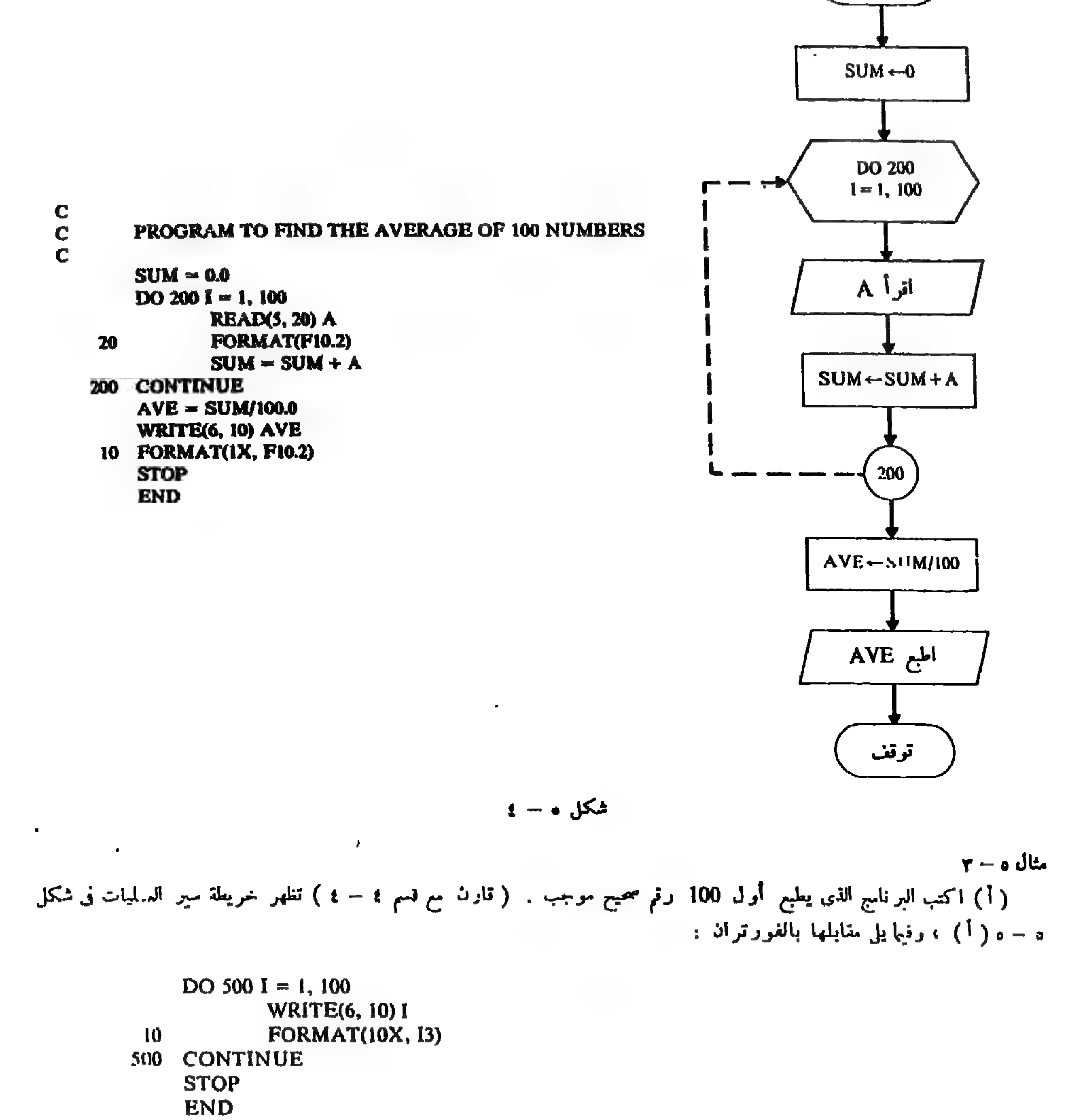

```
      WRITE(6, 5)
    5 FORMAT('1', 3X, 'ID NO.', 4X, 'TEST 1',
     1      4X, 'TEST 2', 4X, 'TEST 3', 3X, 'AVERAGE')
      DO 200 I = 1, 100
            READ(5, 10) ID, S1, S2, S3
   10       FORMAT(5X, I5, 3(4X, F6.2))
            SUM = S1 + S2 + S3
            AVE = SUM/3.0
            WRITE(6, 20) ID, S1, S2, S3, AVE
   20       FORMAT(5X, I5, 4(4X, F6.2))
  200 CONTINUE
      STOP
      END
```

شكل ٥ — ٣

مثال ٥ — ٢

فى قسم ٤ — ١٠ من الفصل الرابع ، كتبنا برنامجاً لإيجاد متوسط 100 رقم . وسنعيد كتابة هذا البرنامج باستعمال الصورة البسيطة لحلقة DO .

لاحظ أن الدليل I لحلقة DO فى المثال السابق (الأعلى) يمثل عداداً يعد عدد المرات التى مررنا فيها خلال الحلقة . فى الواقع ، تكون الحلقة DO أكثر قوة إذا كانت قيمة الدليل I تستخدم أيضاً لحساب بداخل حلقة DO ونوضح هذا فى المثال التالى :




```
C
C       PROGRAM TO FIND THE AVERAGE OF 100 NUMBERS
C
        SUM = 0.0
        DO 200 I = 1, 100
              READ(5, 20) A
   20         FORMAT(F10.2)
              SUM = SUM + A
  200   CONTINUE
        AVE = SUM/100.0
        WRITE(6, 10) AVE
   10   FORMAT(1X, F10.2)
        STOP
        END
```

شكل ٥ — ٤

مثال ٥ — ٣

(أ) اكتب البرنامج الذي يطبع أول 100 رقم صحيح موجب . (قارن مع قسم ٤ — ٤) تظهر خريطة سير العمليات في شكل ٥ — ٥ (أ) ، وفيما يلي مقابلها بالفورتران :

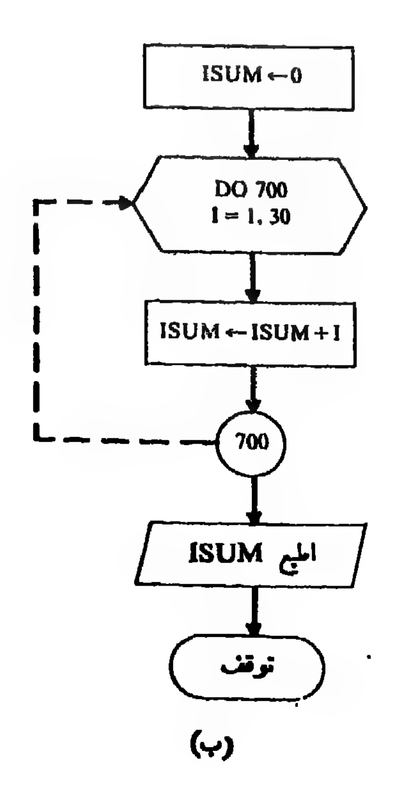
```
      DO 500 I = 1, 100
            WRITE(6, 10) I
   10       FORMAT(10X, I3)
  500 CONTINUE
      STOP
      END
```

(ب) اكتب البرنامج الذي يحسب مجموع أول 30 رقم صحيح موجب . نريد أن نحسب 1 + 2 + 3 + ... + 30 أولا نجعل قيمة ISUM = 0 ثم ننفذ

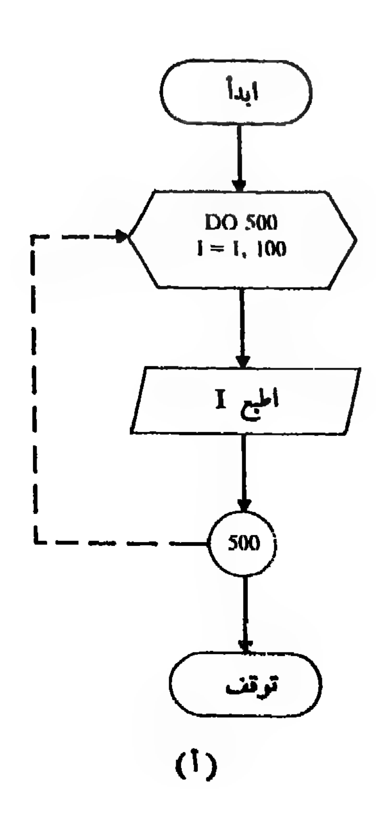
ISUM = ISUM + I

لكل من I = 1, 2, 3, ..., 30 باستعمال حلقة DO التكرارية ، تظهر خريطة سير العمليات في شكل ٥ — ٥ (ب) ، وفيما يلي ما يقابلها بالفورتران :

```
      ISUM = 0
      DO 700 I = 1, 30
            ISUM = ISUM + I
700   CONTINUE
      WRITE(6, 20) ISUM
 20   FORMAT(10X, I10)
      STOP
      END
```



(ب)



(أ)

شكل ٥ – ٥

(جـ) اكتب جزءاً من برنامج يحسب حاصل ضرب أول 10 أرقام صحيحة موجبة . يشبه البرنامج إلى حد كبير ذلك البرنامج في (ب) إلا أننا نريد أن نحسب

1·2·3· . . . ·10

يتم هذا أولا بجعل قيمة IPROD = 1 وتنفذ

IPROD = IPROD*I

باستعمال حلقة DO التكرارية بدليل I وفيما يلي جزء البرنامج المطلوب .

```
      IPROD = 1
      DO 100 I = 1, 10
            IPROD = IPROD*I
100   CONTINUE
```

٥ — ٤ جملة DO

حلقة DO التكرارية التي تمت مناقشتها في القسم السابق هي حالة خاصة فقط . وتأخذ حلقة DO التكرارية (غير الميكيلية) العامة الشكل التالي :

```
DO n I = IN, IE, IC
      :
   جسم الحلقة التكرارية
      :
n  CONTINUE
```

ومعاني الرموز *n* و I و IN و IE و IC هي كالتالي :

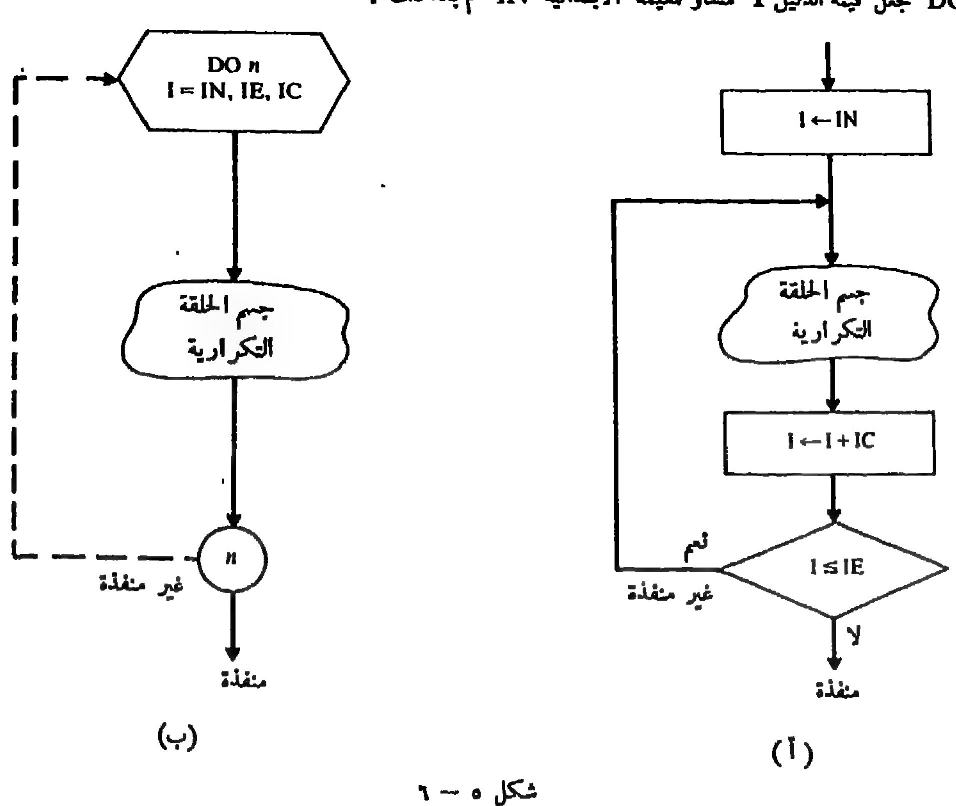
١ — *n* هي رقم آخر جملة قابلة للتنفيذ في حلقة DO التكرارية . وهي جملة CONTINUE في هذه الحالة ، (انظر قسم ٥—٨) .

٢ — تشير I إلى اسم متغير صحيح ، وتسمى دليل الحلقة التكرارية .

٣ — يمكن لكل من IN ، IE ، IC إما أن تكون ثوابت صحيحة موجبة فقط أو أسماء متغيرات صحيحة . ونطلق على هؤلاء معاملات الدليل أو معاملات الحلقة التكرارية. تشير IN إلى القيمة الابتدائية للدليل، IE هي قيمة الاختبار أو قيمة النهاية ، أو القيمة الحدية للدليل ، IC مقدار الزيادة .

ملاحظة : يضم الفورتران الميكل ، الذي يناقش ، في الفصل الثاني عشر ، أنواع أخرى من حلقات DO التكرارية . أحد هذه الأنواع يشبه الحلقة التكرارية السابقة إلا أنها تسمح بمقدار زيادة سالب وتسمح بقيم ابتدائية وقيم نهائية غير موجبة .

تبين خريطة سير العمليات في شكل ٥ — ٦ (أ) حلقة DO التكرارية السابقة (انظر أيضاً الشكل ٥ — ٢) . عندما تقابل جملة DO تجعل قيمة الدليل I مساو للقيمة الابتدائية IN ثم بعد ذلك .



(أ) (ب)

شكل ٥ — ٦

يمر التحكم خلال جسم حلقة DO التكرارية ، أى ، من أول جملة تلى جملة DO حتى جملة CONTINUE . يزاد الآن قيمة الدليل I بمقدار معامل الزيادة IC وتختبر قيمة I الجديدة . إذا تجاوزت I قيمة النهاية IE ، حينئذ تكون حلقة DO التكرارية قد تم تنفيذها : ويمر التحكم إلى أول جملة تلى حلقة DO التكرارية (أى إلى أول جملة بعد جملة CONTINUE) . وإلا فسوف ينفذ جسم حلقة DO التكرارية مرة ثانية ، وهكذا يأخذ الدليل I قيمة جديدة فى كل مرة يمر الحاسب خلال حلقة DO التكرارية يبين الشكل ٥ – ٦ (ب) خريطة سير العمليات لهذه الحلقة التكرارية ولكن باستعمال الصندوق المطول لحلقة DO التكرارية .

مثال ٥ – ٤

(أ) افرض أن الحاسب قد صادف جملة DO التالية :

```
DO 200 K = 2, 10, 3
```

فيمكن للحاسب أن يدور خلال حلقة DO التكرارية ثلاث مرات ، أولا عندما K = 2 ثم عندما K = 2 + 3 = 5 وثالثاً عندما K = 5 + 3 = 8 . بعد الانتهاء من الدورات الثلاث تزداد قيمة K إلى 11 = 3 + 8 ، وفى هذه الحالة تتجاوز قيمة K القيمة النهائية 10 ومن ثم فلا يدور الحاسب خلال حلقة DO التكرارية مرة رابعة . وهذا يوضح أنه ليس من الضرورى أن يساوى الدليل قيمة الاختبار (القيمة الحدية) .

(ب) افرض أننا نريد أن نطبع الأعداد الصحيحة الزوجية بين 2 و 100 أى 100 ,...... ,6 ,4 ,2 . نجعل فى البداية IEVEN = 2 وننفذ IEVEN = IEVEN + 2 عدة مرات طالما IEVEN ≤ 100 . وتظهر خريطة سير العمليات فى شكل ٥ – ٧ (أ) . وباستعمال حلقة DO التكرارية تنتج خريطة سير العمليات فى شكل ٥ – ٧ (ب) ، والمبينة فيما يلى :

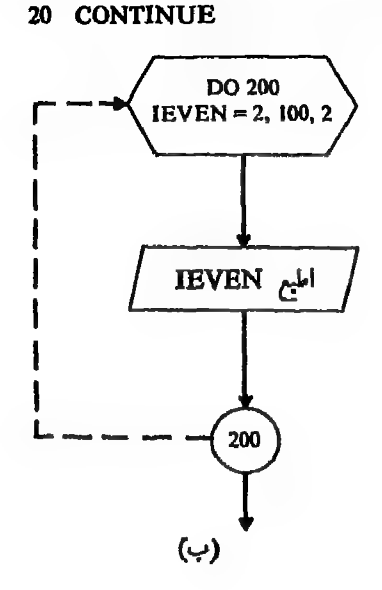
```
      DO 200 IEVEN = 2, 100, 2
            WRITE(6, 10) IEVEN
   10       FORMAT(6X, I3)
  200 CONTINUE
```

(جـ) افرض أن A تم تخزينها فى الذاكرة . احسب حواصل الضرب .

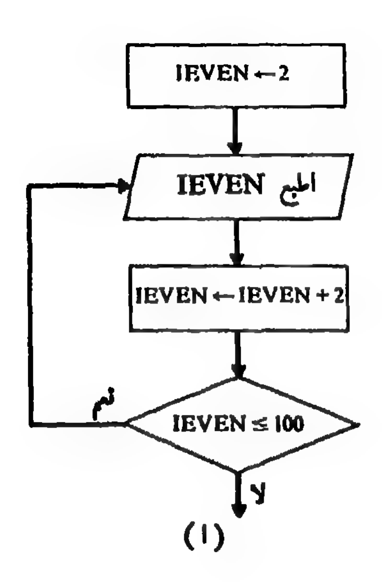
$$(1 + A)(1 + 2A)(1 + 3A)\dots(1 + 10A)$$
$$(1 + A)(1 + 4A)(1 + 7A)\dots(1 + 19A)$$

وكما فى مثال ٥ – ٣ (جـ) فيمكن أن يحسب حاصل الضرب (i) باستعمال حلقة DO التكرارية كما يلى :

```
      PROD = 1.0
      DO 20 I = 1, 10
            PROD = PROD*(1.0 + FLOAT(I)*A)
   20 CONTINUE
```



(ب)



(أ)

شكل ٥ – ٧

وبتعديل طفيف يمكن أن يحسب حاصل الضرب (ii) بالمثل كما يلي :

```
      PROD = 1.0
      DO 50 I = 1, 19, 3
         PROD = PROD*(1.0 + FLOAT(I)*A)
50    CONTINUE
```

٥ — ٥ قوانين على استخدام حلقة DO التكرارية

فيما يلي عدد من القوانين تتعلق بحلقة DO التكرارية غير الهيكلية السابقة .

١ – إذا كان معامل الزيادة 1 يمكن أن يحذف كلية وعلى سبيل المثال

```
DO 100 I = 1, N, 1
```

يمكن أن تكتب في الشكل المختصر

```
DO 100 I = 1, N
```

٢ – يجب أن تكون كل من معاملات الفهرسة IN ، IE ، IC في أى جملة DO ثابت صحيح موجب ، أو متغير صحيح بدون دليل لقيمة موجبة ، وبذلك فإن

```
DO 200 I = 20, 1, -2
```

غير صحيحة (انظر ملاحظة على صفحة ١٤٠)

تستحق النقطتان التاليتان اهتماماً خاصاً حيث أنهما يقودان إلى أخطاء دائمة .

٣ – لا يمكن القيام بحسابات رياضية في جملة DO نفسها . وبذلك فإن

```
DO 300 I = M, K + 3, 2
```

غير صحيحة ، بينما ، يمكن الفرد أن ينجز نفس الحسابات الرياضية بإدخال متغير جديد ، وليكن KK وكتابة .

```
KK = K + 3
DO 300 I = M, KK, 2
```

٤ – رغم أن الدليل I متاح للحسابات داخل حلقة DO التكرارية لا يجب تغييره بداخل جسم حلقة DO التكرارية . وهذا ينطبق أيضاً بالنسبة إلى معاملات الفهرسة IN ، IE ، IC وبمعنى آخر ، لا يمكن تغيير أى من القيم I ، IN ، IE ، IC بداخل حلقة DO التكرارية (إلا عن طريق ميكانيكية التحكم الملازمة)

مثال ٥ — ٥

(أ) ادرس جملة DO التالية

```
DO 200 I = 4, 2, 3
```

لاحظ أن القيمة الابتدائية IN تتجاوز القيمة النهائية IE ومع ذلك . ستنفذ حلقة DO التكرارية مرة واحدة ، تبعاً لخريطة سير العمليات (ولكن في بعض المترجمات ، قد يسبب رسالة خطأ)

(ب) افرض أننا نريد أن نطبع الأعداد الصحيحة 100 و 99 و 98 و .... 1 بهذا الترتيب . فجزء البرنامج التالى الذى يستعمل حلقة DO التكرارية غير الهيكلية السابق استخدامها ليس صحيحاً حيث لا يمكن أن يكون معامل الزيادة سالباً .

```
      DO 500 I = 100, 1, -1
             WRITE(6, 20) I
  20         FORMAT(1X, I3)
 500  CONTINUE
```

ومع ذلك ، فما زالت أمامنا فرصة أن نكتب مثل هذا البرنامج باستعمال حلقة DO التكرارية غير الهيكلية . ونلاحظ أولا أننا يمكن أن نستخدم الجملة .

```
      DO 500 I = 1, 100
```

بينما نريد أن نطبع 100 عندما تكون I هى 1 ونريد أن نطبع 99 عندما 2 = I وهكذا . أى

قيمة I :	1 ، 2 ، 3..... ، 99 ، 100
القيمة التى يجب طباعتها :	100 ، 99 ، 98 ...... ، 2 ، 1

ويجعل J تشير إلى القيمة التى نريد طبعها ، لاحظ أن I + J تساوى دائماً 101 ومن ثم J = 101 — I وبذلك فجزء البرنامج التالى ، الذى يستخدم حلقة DO التكرارية غير الهيكلية ، سوف ينجز مهمتنا .

```
      DO 500 I = 1, 100
             J = 101 - I
             WRITE(6, 20) J
  20         FORMAT(1X, I3)
 500  CONTINUE
```

٥ ـــ ٦ الخروج من حلقة DO التكرارية

هناك طريقتان للخروج من أى حلقة تكرارية DO :
(أ) مخرج طبيعى ، (ب) مخرج غير طبيعى . وسوف نناقش ذلك فيما يلى :

(أ) مخرج طبيعى

يحدث المخرج الطبيعى عندما تتجاوز قيمة الدليل I قيمة الاختبار IE (كما هو موضح فى خريطة سير العمليات فى شكل ٥ ـ ٦ (أ) . فى هذه الحالة ينتقل التحكم إلى أول جملة قابلة التنفيذ تلى حلقة DO التكرارية صفة واحدة خفية للمخرج الطبيعى من حلقة DO التكرارية هو أن قيمة الدليل I عند وقت الخروج تكون غير معرفة ، ومن ثم لا يجب أن تستعمل فى أى حسابات أخرى . كل الأمثلة التى تمت مناقشتها حتى الآن لها مخرج طبيعى .

(ب) مخرج غير طبيعى

من الممكن الانتقال من داخل إلى خارج حلقة DO التكرارية ، ولكن بواسطة جملة IF فى حلقة DO التكرارية ، حتى إذا لم تتجاوز قيمة الدليل I الحالية قيمة الاختبار IE . سيطلق على مثل هذا المخرج مخرج غير طبيعى . صفة واحدة جوهرية للمخرج غير الطبيعى من حلقة DO التكرارية هو أننا نحتفظ بقيمة الدليل I الحالية عند وقت الخروج (من ثم يمكن أن تستخدم هذه القيمة فى أى حسابات أخرى أو فى عمليات I/O)

يوضح هيكل البرنامج التالى ، والذى يستخدم K كدليل لحلقة DO التكرارية متى تكون قيمة الدليل K ، معرفة أو غير معرفة .

```
      DO 100 K = 1, 100, 2
         ***
         ***
         IF(X.LT.Y) GO TO 200
         ***
         ***
100   CONTINUE
      ***
      ***     قيمة الدليل K غير معرفة بالنسبة لهذه الجمل
      ***
      STOP
200   ***
      ***     قيمة الدليل K معرفة بالنسبة لهذه الجمل
      ***
      STOP
```

مثال ٥ — ٦

اكتب البرنامج الذى يقرأ عدداً صحيحاً $K > 2$ ويحدد ما إذا كانت K عدداً أولياً أم لا . إن لم تكن ، أعرض المقسوم عليه غير العادى ، لـ K (قارن مع مثال ٤ — ٥) لقد برهنا إن لم تكن K عدداً أولياً فإن K لها مقسوم عليه غير عادى $\geq K/2$. تظهر خريطة سير العمليات والبرنامج فى شكل ٥ — ٨ . ونلاحظ أن دليل الحلقة التكرارية يتغير من 2 حتى $K/2$ وحيث لا يمكن إجراء أى حسابات فى جملة DO فنستخدم الجملة

أولا $KK = K/2$

لاحظ أن حلقة DO التكرارية تختبر كلا من الأعداد الصحيحة 2 ، 3 ، ..... KK لمعرفة إذا كان العدد يقبل القسمة على K إن لم يكن ، تستكمل الحلقة التكرارية وبذلك يحدث المخرج الطبيعى عندما لا يكون أى من الأعداد 2 ، 3 ، .... KK قاسماً لـ K وبذلك يكون K عدداً أولياً . بينما أى وقت تقبل I القسمة على K يكون لدينا مخرج غير طبيعى . فى هذه الحالة ، لا تزال قيمة الدليل I معرفة ومن ثم يمكن تنفيذ

٥ — ٧ الانتقال بداخل وإلى حلقة DO تكرارية

يمكن دائماً أن ننتقل من أى نقطة فى حلقة DO التكرارية إلى نقطة أخرى فى نفس الحلقة التكرارية ، ويمكن أن نقفز بحرية إلى خارج أى حلقة DO التكرارية ، إلا أنه ليس ممكناً أن نقفز إلى منتصف حلقة تكرارية ، أى أن الطريقة الوحيدة للانتقال إلى جمل بداخل حلقة DO التكرارية هى عن طريق جملة DO الأصلية . من المهم أن نتذكر أن فى كل مرة ينتقل التحكم إلى جملة DO يعاد قيمة دليل الحلقة التكرارية إلى قيمتها الابتدائية .

افرض أننا نريد جزء برنامج فورتران يستخدم حلقة (DO) تكرارية التى تحسب المجموع

$$1 + 2 + 3 + 5 + 6 + 7 + 8 + 9 + 10$$

أى مجموع أول 10 أرقام صحيحة موجبة باستثناء العدد الصحيح 4 ادرس جزئى البرنامجين التاليين :

(أ)

```
      ISUM = 0
      DO 100 I = 1, 10
         IF(I.EQ.4) I = I + 1
         ISUM = ISUM + I
  100 CONTINUE
```

(ب)

```
      ISUM = 0
   50 DO 200 I = 1, 10
         IF(I.EQ.4) GO TO 50
         ISUM = ISUM + I
  200 CONTINUE
```

```
      READ(5, 8) K
    8 FORMAT(I10)
      KK = K/2
      DO 100 I = 2, KK
         IF(K.EQ.(K/I)*I) GO TO 50
  100 CONTINUE
      WRITE(6, 10) K
   10 FORMAT(10X, I5, 1X, 'IS A PRIME')
      STOP
   50 WRITE(6, 20) K, I
   20 FORMAT(10X, I5, 1X, 'IS NOT A PRIME'/
     1       10X, I5, 1X, 'IS A DIVISOR')
      STOP
      END
```


شكل ٥ — ٨

البرنامج (أ) غير صحيح حيث تم تغيير قيمة الدليل I فى حلقة DO التكرارية أى فى الجمل

```
IF(I.EQ.4) I = I + 1
ISUM = ISUM + I
```

إن لم تكن موجودة فى حلقة DO التكرارية ، إذن فستضاف 5 إلى ISUM إذا كانت I تحتوى أصلا على 4 . ومع ذلك حيث أننا هنا فى منتصف حلقة DO التكرارية فغير مسموح بهذه الجمل .

وعلى النقيض فالبرنامج (ب) يستخدم حلقة DO التكرارية بطريقة صحيحة . ومع ذلك فمنطق البرنامج ليس صحيحاً . أى

يوجد مخرج غير طبيعي حين تأخذ I القيمة 4 لكن ينتقل التحكم إلى الخلف أى إلى جملة DO وكما ذكرنا من قبل ، كلما أعدنا تنفيذ جملة DO فتعاد قيمة الدليل إلى قيمته الابتدائية التى هى فى هذه الحالة 1 وبالتالى سيعطى هذا البرنامج حسابات تكرارية لا نهائية .

1 + 2 + 3 + 1 + 2 + 3 + 1 + 2 + 3 + ...

وهكذا

الغرض من البرنامج (ب) هو تخطى ISUM = ISUM + I عندما تكون I تساوى 4 إلا أننا نريد أيضاً أن نكمل حلقة DO التكرارية فى مسارها الطبيعى . ويمكن أن ننجز ذلك بنقل التحكم إلى جملة CONTINUE بدلا من جملة DO أى أن برنامج (ب) سيحسب المجموع المطلوب إذا تغيرت الجملة

```
IF(I.EQ.4) GO TO 50
```

الى

```
IF(I.EQ.4) GO TO 200
```

يوضح هذا المثال حقيقة النقطة الأساسية فى القسم التالى — وهى ضرورة جملة CONTINUE

٥ — ٨ ضرورة جملة CONTINUE

فى الحقيقة ليس من الضرورى أن تكون جملة CONTINUE هى الجملة الأخيرة فى أى حلقة تكرارية DO ، ولكن يجب أن تكون أى جملة قابلة للتنفيذ غير جملة GO TO أو جملة IF الحسابية ، أو جملة DO أخرى . على سبيل المثال فبدلا من

```
      ISUM = 0
      DO 200 I = 1, 100
         ISUM = ISUM + I
200   CONTINUE
```

يمكن أن نكتب

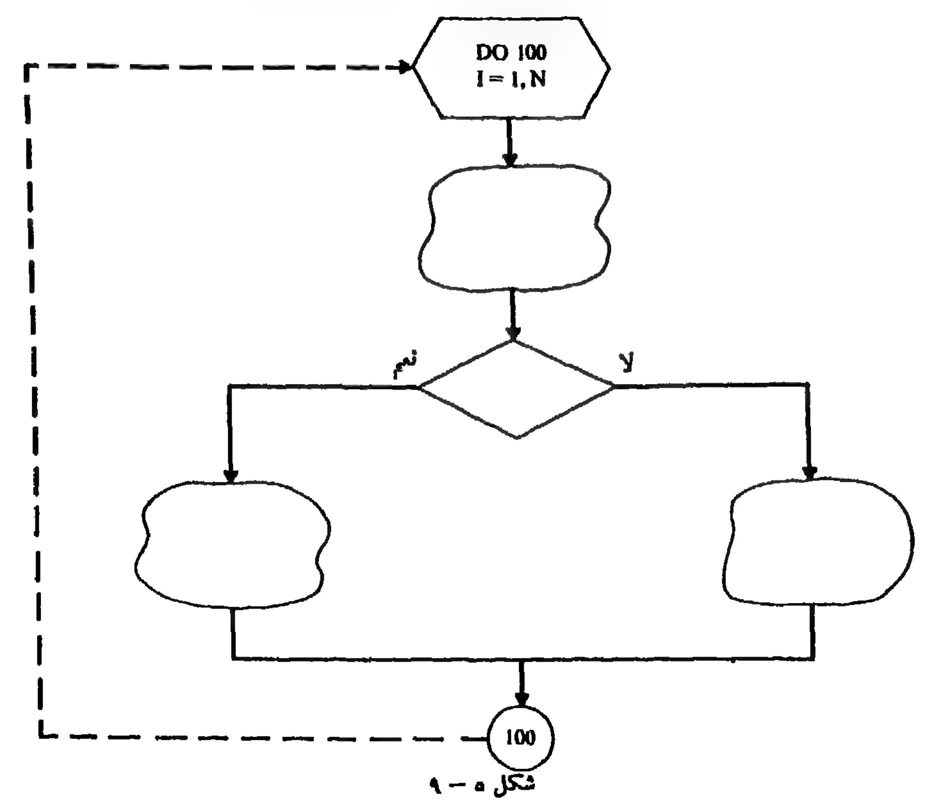
```
      ISUM = 0
      DO 200 I = 1, 100
200      ISUM = ISUM + I
```

ومع ذلك فما زلنا نحبذ استخدام الزوج DO-CONTINUE إذ أنها ستمنع نوع الخطأ الشائع التالى :

```
      DO 300 I = 1, 100
         WRITE(6, 300) I
300      FORMAT(10X, I3)
```

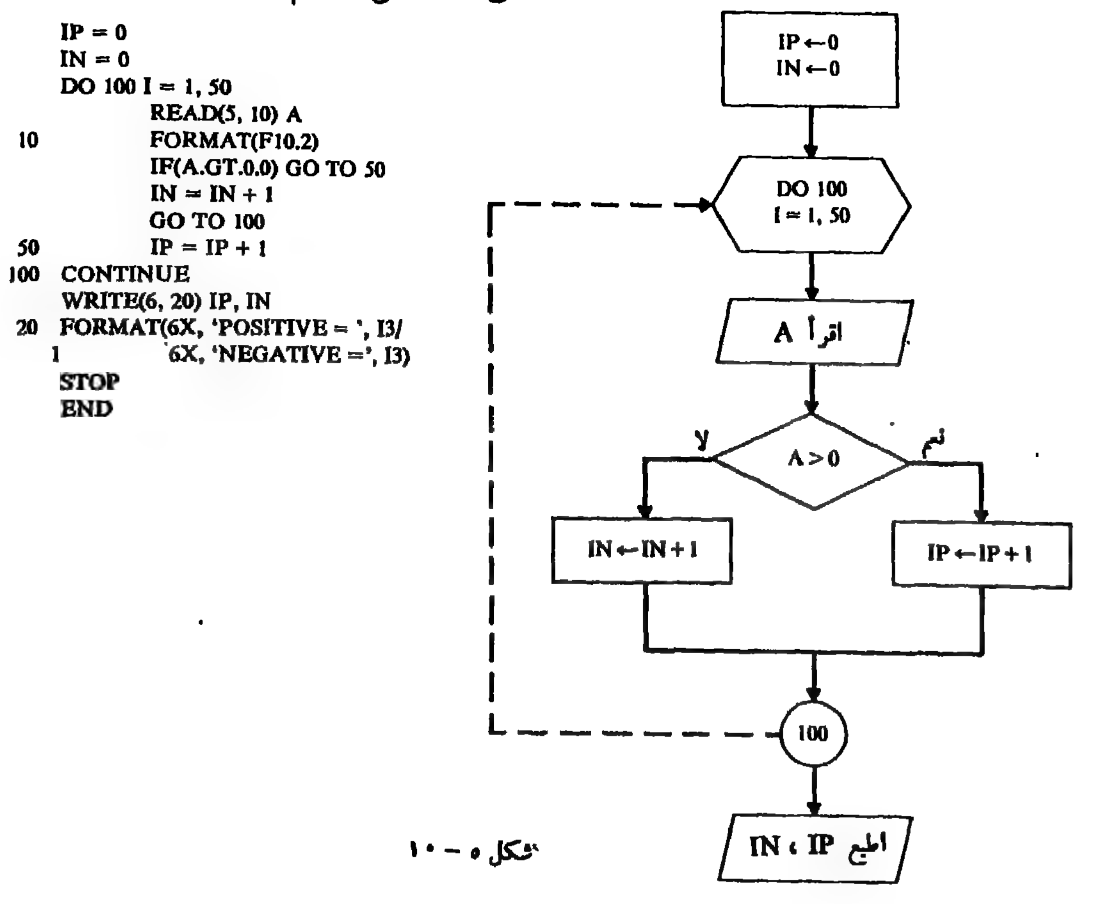
(الخطأ هو استعمال جملة FORMAT غير المنفذة كآخر جملة فى الحلقة التكرارية DO فجملة WRITE بدلا من جملة FORMAT هى التى يجب أن تحمل الرقم 300. وأكثر من ذلك أهمية ، يعرض الزوج DO-CONTINUE الميزة التعليمية حيث أنه يقفل شكل أى حلقة تكرارية DO ، وجملة CONTINUE تخدم كحدود للحلقة التكرارية DO .

ومع ذلك ، هناك موقف يجب أن نستخدم فيه جملة CONTINUE افرض أن هناك جملة شرطية بداخل الحلقة التكرارية DO حيث البديلان ليس بينهما شىء عام ، كما هو موضح فى الشكل ٥ — ٩ .



شكل ٥ – ٩

مطلوب نوع خاص من الجمل القابلة للتنفيذ يمكن للبديلين أن يشيرا إليها لتمكن الحلقة التكرارية DO من استكمال مسارها الطبيعي وقد خلقت جملة CONTINUE لهذا الغرض ومثال ذلك برنامج ب المصحح في قسم ٥ – ٧ .



```
        IP = 0
        IN = 0
        DO 100 I = 1, 50
              READ(5, 10) A
 10           FORMAT(F10.2)
              IF(A.GT.0.0) GO TO 50
              IN = IN + 1
              GO TO 100
 50           IP = IP + 1
100     CONTINUE
        WRITE(6, 20) IP, IN
 20     FORMAT(6X, 'POSITIVE = ', I3/
      1        6X, 'NEGATIVE =', I3)
        STOP
        END
```


شكل ٥ – ١٠

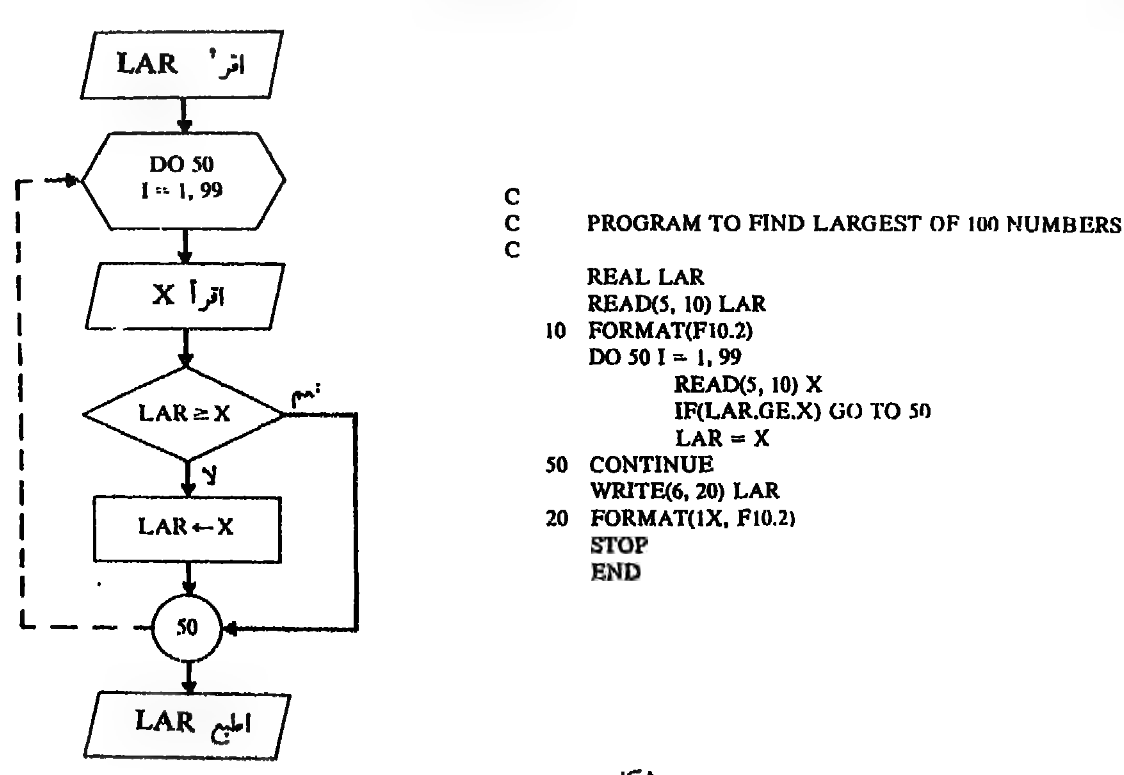
مثال ٥ — ٧

أعطيت مجموعة من 50 بطاقة . ثقب على كل بطاقة رقم غير الصفر . أكتب برنامج الفورتران الذي يعد عدد الأرقام الموجبة وعدد الأرقام السالبة .

تظهر خريطة سير العمليات للبرنامج وما يقابله بالفورتران فى شكل ٥ — ١٠ وقد اضطررنا هنا للمرة الثانية أن نستخدم جملة CONTINUE

مثال ٥ — ٨

اكتب برنامجاً لإيجاد أكبر رقم من 100 رقم ، حيث تم تثقيب كل رقم على بطاقة منفصلة . تظهر خريطة سير العمليات وما يقابلها بالفورتران فى شكل ٥ — ١١. لاحظ أن القيمة النهائية لجملة DO هى 99 وليس 100 وهذا نظراً لأن أول رقم يقرأ يخزن فى LAR قبل تنفيذ حلقة DO التكرارية . مرة أخرى ، يجب أن نستخدم جملة CONTINUE




```
C
C       PROGRAM TO FIND LARGEST OF 100 NUMBERS
C
        REAL LAR
        READ(5, 10) LAR
   10   FORMAT(F10.2)
        DO 50 I = 1, 99
              READ(5, 10) X
              IF(LAR.GE.X) GO TO 50
              LAR = X
   50   CONTINUE
        WRITE(6, 20) LAR
   20   FORMAT(1X, F10.2)
        STOP
        END
```

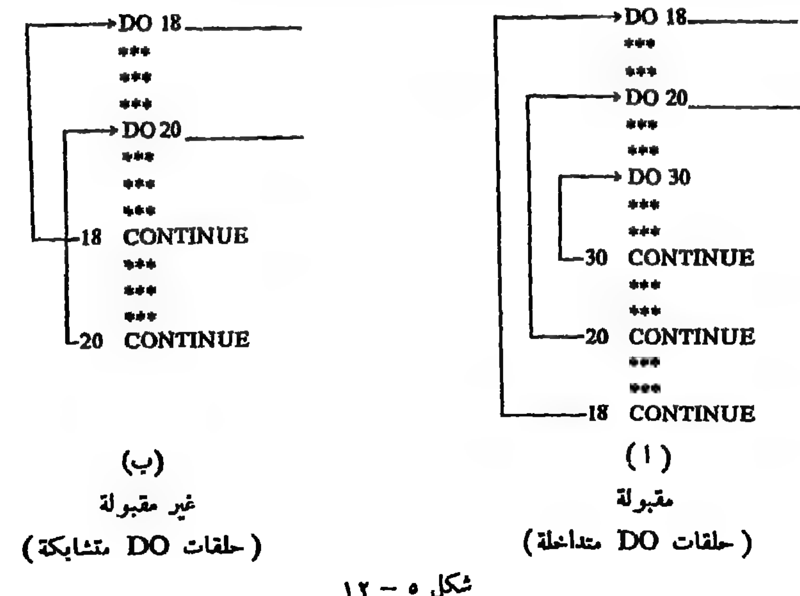
شكل ٥ — ١١

٥ — ٩ حلقات DO التكرارية المتداخلة

من الممكن أن تكون لدينا حلقة DO تكرارية (حلقة DO تكرارية داخلية) . تقع بالكامل داخل مدى حلقة DO تكرارية أخرى (حلقة DO تكرارية خارجية) . تسمى حلقات DO التكرارية التى تظهر بهذا النمط حلقات DO التكرارية المتداخلة والقوانين التى تطبق على حلقـات DO التكرارية المتداخلة أساساً هى نفسها التى تطبق على حلقة DO التكرارية المفردة . بينما تراعى بعض النقاط الهامة التالية :

١ — حيث أن الدليل لا يمكن أن يعاد تعريفه داخل جسم حلقة DO التكرارية ، فدليل حلقة DO التكرارية الداخلية لا يجب أن يكون هو نفسه دليل حلقة DO التكرارية الخارجية .

٢ – يجب أن تقع حلقة DO التكرارية الداخلية تماماً بداخل حلقة DO التكرارية الخارجية ، أى ، لا يجب أن تتشابك الحلقات التكرارية . يوضح شكل ٥ – ١٢ طريقة تداخل حلقات DO التكرارية .



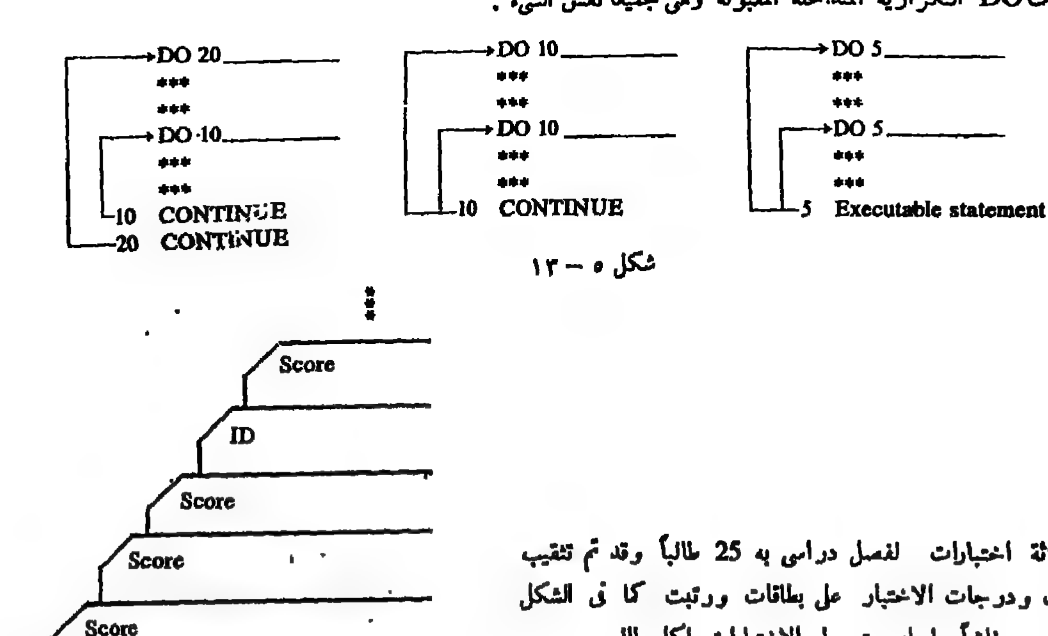
(ا)
مقبولة
(حلقات DO متداخلة)

(ب)
غير مقبولة
(حلقات DO متشابكة)

شكل ٥ – ١٢

٣ – تحدد عدد حلقات DO التكرارية المتداخلة بواسطة المترجم . يمكن أن يكون نقل التحكم بداخل حلقات DO التكرارية المتداخلة بصورة خادعة . ولكن القوانين تظل هى نفسها كما مع حلقات DO التكرارية المفردة .

وأخيراً ، نلاحظ أن حلقات DO التكرارية المتداخلة يمكن أن تكون لها نفس الجملة الأخيرة . ويبين الشكل ٥ – ١٣ حلقات DO التكرارية المتداخلة المقبولة وهى جميعا نفس الشيء .



شكل ٥ – ١٣

شكل ٥ – ١٤

مثال ٥ – ٩

أعطيت ثلاثة اختبارات لفصل دراسى به 25 طالباً وقد تم تثقيب رقم ID الطالب ودرجات الاختبار على بطاقات ورتبت كما فى الشكل ٥ – ١٤ . اكتب برنامجاً لحساب متوسط الاختبارات لكل طالب .

يمكن حساب متوسط درجات الثلاث اختبارات بواسطة جزء البرنامج التالى .

```
      SUM = 0.0
      DO 20 I = 1, 3
            READ(5, 10) SCORE
10          FORMAT(F6.2)
            SUM = SUM + SCORE
20    CONTINUE
      AVE = SUM/3.0
```

وحيث أنه يجب علينا أن نحسب المتوسط لكل من 25 طالب ، فيجب أن نكرر الجزء السابق 25 مرة . لذلك نحيط الجزء السابق بحلقة DO تكرارية أخرى كما يلي :

```
      WRITE(6, 100)
100   FORMAT('1', 4X, 'ID', 9X, 'AVERAGE')
      DO 80 K = 1, 25
            READ(5, 90) ID
90          FORMAT(I10)
            SUM = 0.0                      <--+
            DO 20 I = 1, 3                    |
                  READ(5, 10) SCORE           |
10                FORMAT(F6.2)                |  نفس الجزء السابق
                  SUM = SUM + SCORE           |
20          CONTINUE                          |
            AVE = SUM/3.0                  <--+
            WRITE(6, 30) ID, AVE
30          FORMAT(1X, I10, 5X, F6.2)
80    CONTINUE
```

ملاحظة : لاحظ زحزحة حلقة DO التكرارية الداخلية بداخل حلقة DO التكرارية الخارجية . ومرة أخرى ، يفضل هذا من أجل تسهيل القراءة مع العلم أن هذا لا يؤثر على تشغيل البرنامج .

مسائل محلولة

حلقات DO التكرارية :

٥ — ١ حدد عدد مرات تكرار كل حلقة من حلقات DO التكرارية وبين كذلك لأى قيمة من قيم الدليل سوف تنفذ كل حلقة .

(أ) DO 10 K = 1, 11, 3   (ب) DO 20 JIM = 4, 12

(جـ) DO 30 LARGE = 8, 18, 15   (د) DO 40 M = 7, 4, 2

(أ) القيمة الابتدائية هي 1 ومعامل الزيادة هو 3 ومن ثم ستنفذ حلقة DO التكرارية طالما $K = 1$ و $K = 1 + 3 = 4$ و $K = 4 + 3 = 7$ و $K = 7 + 3 = 10$ وعندما تزداد K المرة التالية ستكون قيمتها أكبر من القيمة النهائية (11) لذا سينتقل التحكم خارج حلقة DO التكرارية . وبذلك ستنفذ حلقة DO التكرارية أربع مرات .

(ب) حيث أن قيمة معامل الزيادة غير موجود تكون قيمة معامل الزيادة 1 ومن ثم ستنفذ حلقة DO التكرارية طالما $JIM = 4, 5, \ldots, 12$ وبذلك تنفذ تسع مرات .

(جـ) تنفذ حلقة DO التكرارية مرة واحدة فقط وذلك عندما تكون $LARGE = 8$ ، وحيث أن القيمة التالية لـ LARGE هي $8 + 15 = 23$ وهي أكبر من قيمة الاختبار 18 فسوف يتوقف تنفيذ الحلقة التكرارية .

(د) رغم أن القيمة M الابتدائية أكبر من القيمة النهائية إلا أن حلقة DO التكرارية سوف تنفذ مرة واحدة عندما تكون $M = 7$ حيث أن الاختبار يتم في نهاية حلقة DO التكرارية (وستعطى بعض المترجمات رسالة خطأ).

٥ – ٢ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل DO :

(أ) DO 100, K = 3, 24, 2
(ب) DO 200 JACK = 5, M, 4
(جـ) DO 300 JILL = JOHN, JIM
(د) DO 400 K = 4, 2*M, 3

(أ) يجب ألا تكون هناك فصلة بعد 100 .

(ب) يجب ألا تكون هناك فصلة بعد 4 .

(جـ) لا توجد هناك أخطاء (مع فرض أن JOHN و JIM قد تم تعريفهما)

(د) لا يمكن أن تتم أى عملية حسابية فى أى من معاملات حلقة DO التكرارية .
يجب أن تتغير الجملة ، وليكن ، إلى الجملتين التاليتين .

```
MM = 2*M
DO 400 K = 4, MM, 3
```

٥ – ٣ أوجد القيمة النهائية لـ K بعد تنفيذ كل جزء من برنامج الفورتران :

(أ)
```
   K = 2
   DO 10 I = 3, 8, 2
      K = K + I
10 CONTINUE
   K = 2*K
```

(ب)
```
   K = 2
   DO 20 I = 3, 8, 2
      K = K + I
      IF(K.GT.6) GO TO 30
20 CONTINUE
30 K = 2*K
```

(جـ)
```
   K   2
   M = 2
   DO 40 I = 3, 8, M
      K = K + I
40 CONTINUE
   K = 2*K
```

(د)
```
   K = 2
   DO 50 I = 3, 8, K
      K = K + I
50 CONTINUE
   K = 2*K
```

(أ) تخصص أول جملة 2 إلى K . ثم تنفذ بعد ذلك حلقة DO التكرارية كما يلى :

١ – أولا عندما I = 3 ينتج

K ← K + I = 2 + 3 = 5

٢ – ثم عندما I = 5 ينتج

K ← K + I = 5 + 5 = 10

٣ – ثم عندما I = 7 ينتج

K ← K + I = 10 + 7 = 17

تتجاوز قيمة I التالية قيمة الاختبار ، لذا ينتقل التحكم إلى الجملة الأخيرة التى تضاعف قيمة K

K ← 2*K = 2·17 = 34

وبذلك تكون قيمة K النهائية هى 34

(ب) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلى :

١ – أولا عندما I = 3 ينتج :

K ← K + I = 2 + 3 = 5

وحيث أن $K < 6$ فلا ينتقل التحكم إلى الجملة التى تحمل الرقم 30

٢ – ثم عندما $I = 5$ ينتج

$$K \leftarrow K + I = 5 + 5 = 10$$

$$K \leftarrow K + I = 5 + 5 = 10$$

وحيث $K > 6$ فسوف ينتقل التحكم إلى الجملة التى تحمل الرقم 30 أى إلى الجملة الأخيرة . تضاعف الجملة الأخيرة قيمة K لتصبح :

$$K \rightarrow 2*K = 2*10 = 20$$

$$K \leftarrow 2*K = 2 \cdot 10 = 20$$

ومن ثم ، تكون قيمة K النهائية هى 20

(جـ) حيث أن $M = 2$ وهذا هو نفس البرنامج كما فى (أ) من ثم فقيمة K الأخيرة هى 34

(د) حيث أن $K = 2$ فجملة DO هى نفسها كما فى (أ) ، ومع ذلك فإن K الآن معامل وقيمتها تتغير فى البرنامج ، وهذا غير مسموح به . وبذلك لا يمكن تنفيذ جزء البرنامج .

٥ – ٤ أوجد قيمة K النهائية بعد تنفيذ كل أجزاء برنامج الفورتران :

(أ)

```
   K = 2
10 DO 20 I = 3, 8, 2
         IF(I.EQ.5) GO TO 20
         K = K + I
20 CONTINUE
   K = 2*K
```

(ب)

```
   K = 2
10 DO 20 I = 3, 8, 2
         IF(I.EQ.5) GO TO 10
         K = K + I
20 CONTINUE
   K = 2*K
```

(أ) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلى :

(١) أولا عندما $I = 3$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التى تعطى

$$K \leftarrow K + I = 2 + 3 = 5$$

(٢) ثم عندما $I = 5$. وحيث $I = 5$ لذا ينتقل التحكم إلى جملة CONTINUE التى تعيد دورة الحلقة التكرارية .

(٣) ثم عندما $I = 7$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التى تعطى

$$K \leftarrow K + I = 5 + 7 = 12$$

تتجاوز قيمة I التالية قيمة الاختبار ، ولذا ينتقل التحكم إلى الجملة التالية لحلقة DO التكرارية ، التى تضاعف K . ومن ثم تكون قيمة K النهائية 24

(ب) تخصص الجملة الأولى 2 إلى K ثم تنفذ حلقة DO التكرارية كما يلى :

(١) أولا عندما $I = 3$ وحيث أن $I \neq 5$ تنفذ الجملة $K = K + I$ التى تعطى

$$K = K + I = 2 + 3 = 5$$

(٢) ثم عندما $I = 5$ وحيث أن $I = 5$ فلذا ينتقل التحكم إلى جملة DO . وحيث أن التحكم انتقل إلى جملة DO فتبدأ حلقة DO التكرارية من الأول مرة أخرى وتجعل $I = 3$ وبذلك تتكرر (١) و (٢) مرة ثانية وثالثة . ويعطى هذا حلقة تكرارية لا نهائية ، ولا توجد قيمة نهائية لـ K ، (اننا نؤكد أنه لو انتقل التحكم إلى جملة DO يعاد الدليل إلى قيمته الابتدائية) .

٥ – ٥ اكتشف الأخطاء ، إن وجدت فى كل برنامج . يمثل السهم المقوس نقل التحكم

(أ)

```
C         FIRST PROGRAM
          .................
          .................
          DO 10 I = 1, 25
          .................
          .................
       10 CONTINUE
          .................
          .................
          END
```

(ب)

```
C         SECOND PROGRAM
          .................
          DO 10 I = 1, 25
          .................
          DO 20 J = 1, 15
          .................
          .................
       10 CONTINUE
          .................
       20 CONTINUE
          .................
          .................
          END
```

(جـ)

```
C         THIRD PROGRAM
          .................
          DO 10 I = 1, 25
          .................
          .................
          DO 20 J = 1, 15
          .................
          .................
       20 CONTINUE
          .................
       10 CONTINUE
          .................
          END
```

(د)

```
C         FOURTH PROGRAM
          .................
          DO 10 I = 1, 25
          .................
          .................
          DO 20 J = 1, 15
          .................
          .................
       20 CONTINUE
          .................
       10 CONTINUE
          .................
          .................
          END
```

(أ) لا يمكن نقل التحكم إلى منتصف حلقة DO تكرارية .

(ب) حلقات DO التكرارية متشابكة .

(جـ) لا يمكن نقل التحكم من حلقة DO خارجية إلى داخل حلقة DO داخلية .

(د) لا توجد أخطاء . يمكن أن تنتقل من حلقة DO داخلية إلى حلقة DO خارجية

برامج :

٥ – ٦ اكتب البرنامج الذى يطبع الأعداد الصحيحة الموجبة من 1 إلى 300 مع طبع كل ثلاثة أرقام فى سطر ، حتى يبدو الخرج كالشكل التالى :

```
 1     2     3
 4     5     6
.................
298   299   300
```

نريد طباعة I ، I + 1 ، I + 2 على كل سطر طالما I = 1, 4, 7.... و $I \leq 300$. ومن ثم . نستخدم حلقة DO التكرارية بدليل I يتغير من 1 إلى 300 ومعامل الزيادة 3 ويبدو البرنامج كما يلى :

```
      DO 100 I = 1, 300, 3
         J = I + 1
         K = I + 2
         WRITE(6, 10) I, J, K
10       FORMAT(1X, 3(I5, 3X))
100   CONTINUE
      END
```

٥ — ٧ اكتب برنامج الفورتران الذى يقرأ عدداً صحيحاً فردياً موجباً N ويحسب (إلى ثلاثة أماكن عشرية) المجموع :

(أ) $1 + 1/2 + 1/3 + \cdots + 1/N$ (جـ) $1 - 1/2 + 1/3 - \cdots + 1/N$

(ب) $1 + 1/3 + 1/5 + \cdots + 1/N$

(أ) أولا اجعل SUM = 0.0 ثم استخدم حلقة DO تكرارية بدليل K حيث K = 1, 2...., N وذلك بلجمع 1/K إلى SUM . وفيما يلى نبين البرنامج

```
      READ(5, 10) N
10    FORMAT(I5)
      SUM = 0.0
      DO 100 K = 1, N
         X = FLOAT(K)
         SUM = SUM + 1.0/X
100   CONTINUE
      WRITE(6, 20) N, SUM
20    FORMAT(1X, I5, 3X, F10.3)
      STOP
      END
```

(ب) البرنامج هو نفسه مثل (أ) فيما عدا أن حلقة DO التكرارية يجب أن تستبدل بالتالى .

```
      DO 100 K =  1, N, 2
```

حيث أننا سوف نستخدم قيم K فقط الفردية .

(جـ) البرنامج هو نفسه مثل (أ) فيما عدا أنه يجب استبدال السطر السادس بما يلى

```
      SUM = SUM + (-1.0)**(K + 1)/X
```

حيث أن الإشارات ستتبادل .

٥ — ٨ افترض أنه تم إيداع مبلغ $2000.00 فى حساب توفير سنة 1977 وافترض أن البنك يدفع 6 فى المائة فائدة مركبة سنويا على الحساب . اكتب البرنامج الذى يطبع YEAR و AMOUNT الحساب حتى سنة 1995 (قارن مع مسألة (٤ — ١١)) .

تذكر أولا أن كل سنة تزداد الـ AMOUNT بمقدار 6 فى المائة .

AMOUNT ← AMOUNT + 0.06*AMOUNT

القانون العام هو

AMOUNT ← AMOUNT + RATE*AMOUNT = AMOUNT(1 + RATE)

حيث (RATE هو سعر الفائدة) . وفيما يلى نبين البرنامج

```
      INTEGER YEAR
      WRITE(6, 10)
10    FORMAT('1', 6X, 'YEAR', 6X, 'AMOUNT'//)
      AMOUNT = 2000.00
      YEAR = 1977
      WRITE(6, 20) YEAR, AMOUNT
20    FORMAT(1X, I10, 3X, '$', F8.2)
      DO 99 YEAR = 1978, 1995
            AMOUNT = AMOUNT + 0.06*AMOUNT
            WRITE(6, 20) YEAR, AMOUNT
99    CONTINUE
      STOP
      END
```

(لاحظ أننا استخدمنا حلقة DO التكرارية بدليل YEAR ابتداء من سنة 1978 حيث أن هذه هي أول سنة تخصص فيها الأرباح)

٥ – ٩ اعتبر أنه تم تثقيب مجموعة من البيانات $x_1, x_2, \ldots, x_n$ على بطاقات ، رقم واحد في كل بطاقة ، وتحتوى المجموعة على بطاقة مقدمة (عنوان – بداية)

(ا) اكتب جزءاً من برنامج لحساب المجاميع التالية :

$$SUM = x_1 + x_2 + \cdots + x_n = \sum_{i=1}^{n} x_i$$

$$SUMSQ = x_1^2 + x_2^2 + \cdots + x_n^2 = \sum_{i=1}^{n} x_i^2$$

(رمز التجميع Σ يناقش فى قسم ٨ – ٩ (ج)) .

(ب) يعرف كل من المتوسط ، التباين والانحراف المعيارى $x_1, x_2, \ldots, x_n$ بالآتى :

$$m = \text{mean} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\text{variance} = \frac{\sum_{i=1}^{n} (x_i - m)^2}{n}$$

$$\text{standard deviation} = \sqrt{\text{variance}}$$

مع ذلك فيمكن أن نكتب التباين كالآتى :

$$\text{variance} = \frac{\sum_{i=1}^{n} x_i^2}{n} - m^2$$

اكتب برنامجاً يحسب المتوسط ، التباين ، والانحراف المعيارى للبيانات التى تم تثقيبها .

(ا) فى البداية ، اجعل SUM = 0.0 و SUMSQ = 0.0 . بعد قراءة N استخدم حلقة DO التكرارية لقراءة قيم X وكذلك لجمع X إلى SUM و X^2 إلى SUMSQ . وفيما يلى نبين جزء البرنامج :

(٤)

```
      SUM = 0.0
      SUMSQ = 0.0
      READ(5, 10) N
10    FORMAT(I5)
      DO 100 I = 1, N
            READ(5, 20) X
20          FORMAT(F10.2)
            SUM = SUM + X
            SUMSQ = SUMSQ + X**2
100   CONTINUE
```

(ب) يمكننا القانون الثانى للتباين من حساب التباين فى نفس الوقت مع المتوسط (بدون استخدام المتغيرات ذات الدليل) لذلك سوف نستخدم جزء البرنامج السابق فى برنامجنا التالى :

```
      REAL MEAN
      SUM = 0.0
      SUMSQ = 0.0
      READ(5, 10) N
10    FORMAT(I5)
      DO 100 I = 1, N
            READ(5, 20) X
20          FORMAT(F10.2)
            SUM = SUM + X
            SUMSQ = SUMSQ + X**2
100   CONTINUE
      XN = FLOAT(N)
      MEAN = SUM/XN
      VAR = SUMSQ/XN - MEAN**2
      SD = SQRT(VAR)
      WRITE(6, 30) MEAN, VAR, SD
30    FORMAT(1X, 3(F10.2, 2X))
      STOP
      END
```

٥ — ١٠ طريقة مجموع الأرقام هى إحدى طرق حساب قيمة الاستهلاك . فعلى سبيل المثال ، افترض أن سيارة سعرها $4500.00 ستستهلك قيمتها على مدى خمس سنوات بمجموع أرقام السنوات SUM فى هذه الحالة هو

$$SUM = 1 + 2 + 3 + 4 + 5 = 15$$

إذا طبقنا الطريقة فإن 5/15 من 4500 $ تستهلك فى السنة الأولى ، 4/15 فى السنة الثانية ، 3/15 فى السنة الثالثة ، وهكذا ، يمكن أن نجدول هذه البيانات كالآتى :

السنة	قيمة الاستهلاك
1	1500.00
2	1200.00
3	900.00
4	600.00
5	300.00

اكتب البرنامج الذى يتقبل عدداً موجباً حقيقياً COST وعدداً صحيحاً موجباً N ويطبع قيمة الاستهلاك COST على مدى عدد N من السنوات باستخدام طريقة مجموع الأرقام حتى يبدو الخرج فى الصورة السابقة .

سنستخدم القانون :

$$SUM = 1 + 2 + 3 + \cdots + n = n(n + 1)/2$$

لاحظ أن n/SUM من COST تستهلك أول سنة ، $(n-1)/SUM$ السنة الثانية ، وهكذا ، وبالتحديد $(n-K+1)/SUM$ من COST يستهلك فى السنة K . نستخدم فى البرنامج حلقة DO التكرارية بدليل K = 1, 2, ...N وتحسب قيمة الاستهلاك باستخدام .

```
DEP = FLOAT(N - K + 1)/SUM*COST
```

وفيما يلى البرنامج :

```
C
C        PROGRAM USING SUM-OF-DIGITS DEPRECIATION
C
         READ(5, 10) COST, N
      10 FORMAT(F15.2, I5)
         SUM = N*(N + 1)/2
         WRITE(6, 20)
      20 FORMAT('1', 5X, 'YEAR', 5X, 'DEPRECIATION')
         DO 100 K = 1, N
                DEP = FLOAT(N - K + 1)/SUM*COST
                WRITE(6, 30) K, DEP
      30        FORMAT(6X, I3, 6X, F10.2)
     100 CONTINUE
         STOP
         END
```

٥ - ١١ (أ) ادرس متعددة الحدود التربيعية $y = 2x^2 - 3x - 5$ اكتب برنامج الفورتران الذى يجد y لقيم x والتى تتراوح ما بين 4 — و 4 بخطوات 0.5 .

(ب) ادرس متعددة الحدود .

$$z = x^3 - 3xy^2 + 2xy + y - 2y^3$$

اكتب برنامج فورتران باستخدام حلقة DO التكرارية المتداخلة التى تجد z لقيم x وتأخذ y القيم من 4 — إلى 4 بخطوات 0.5 .

(أ) حيث أن هناك 17 قيمة لـ x لذا سنستخدم حلقة DO التكرارية بدليل I = 1, 2..., 17 . العلاقة بين الدليل I و x هى كالتالى :

x:	−4,	−3.5,	−3,	−2.5,	....,	3.5,	4
I:	1,	2,	3,	4,	....,	16,	17

لاحظ أننا نستطيع أن نحصل على I من قيم x باستخدام

$$x = -4 + 0.5(I - 1)$$

وفيما يلى البرنامج :

```
C       PROGRAM QUADRATIC POLYNOMIAL
        DO 100 I = 1, 17
              X = -4.0 + 0.5*FLOAT(I - 1)
              Y = 2.0*X**2 - 3.0*X - 5.0
              WRITE(6, 10) X, Y
   10         FORMAT(1X, F10.3, 3X, F10.3)
  100   CONTINUE
        STOP
        END
```

(ب) يشابه هذا البرنامج ذلك الموجود فى (ا) فيما عدا أننا الآن لدينا حلقة DO تكرارية لـ x وكذلك حلقة أخرى لـ y

```
C       PROGRAM POLYNOMIAL
        DO 200 I = 1, 17
              X = -4.0 + 0.5*FLOAT(I - 1)
              DO 100 J = 1, 17
                    Y = -4.0 + 0.5*FLOAT(J - 1)
                    Z = X**3 - 3.0*X*Y**2 + 2.0*X*Y + Y - 2.0*Y**3
                    WRITE(6, 10) X, Y, Z
   10               FORMAT(1X, 3(3X, F10.2))
  100         CONTINUE
  200   CONTINUE
        STOP
        END
```

• ١٢ – الرسم البيانى للمعادلة $x^2 + y^2 = 50$ هو دائرة C مركزها عند نقطة الأصل ونصف قطرها $\sqrt{50}$

(ا) حدد عدد النقط ذات الإحداثيات الصحيحة الموجبة التى تقع فى الدائرة .

(ب) حدد عدد النقط ذات الإحداثيات الصحيحة التى تقع فى الدائرة .

(ا) لاحظ أولا أن قيم x و y لا يمكن أن تتجاوز 7 حيث أن نصف القطر هو $\sqrt{50}$. نستخدم حلقتين من حلقات DO التكرارية المتداخلة واحدة لـ x وواحدة لـ y . نبدأ أيضاً بعداد K باعطاءه قيمة ابتدائية 0 ولا نحسب تلك النقط (x, y) بحيث تكون $x^2 + y^2 \geq 50$. وفيما يلى ذلك البرنامج . لاحظ أن x و y قد تم تعريفها على أنهما متغيرات صحيحة .

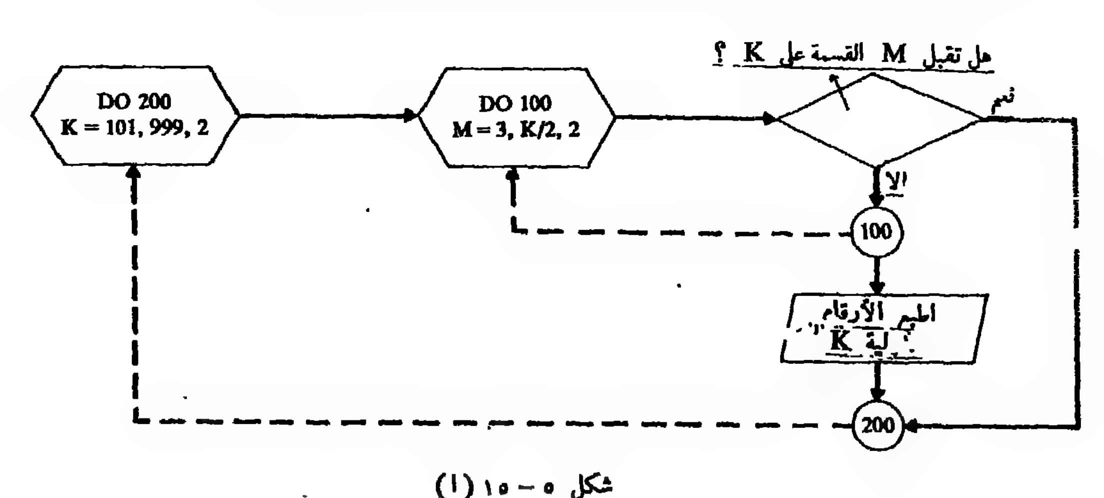
```
C       PROGRAM COUNTING POINTS IN CIRCLE
        INTEGER X, Y
        K = 0
        DO 100 X = 1, 7
              DO 200 Y = 1, 7
                    IF(X**2 + Y**2.GE.50) GO TO 200
                    K = K + 1
  200         CONTINUE
  100   CONTINUE
        WRITE(6, 10) K
   10   FORMAT(1X, 'THE NUMBER OF POINTS IS', 2X, I5)
        STOP
        END
```

(ب) يجب أن نختبر هنا 15 قيمة لـ x و y أى $-7, -6, \ldots\ldots, 7$ ، ومن ثم ، جهز حلقة DO تكرارية بدليل $I = 1, 2, \ldots.. 15$ لـ x حيث نجعل $x = 8 - I$ وحلقة DO تكرارية مثيلة لـ y وفيما عدا ذلك فالبرنامج يشابه الجزء (ا) .

```
      INTEGER X, Y
      K = 0
      DO 100 I = 1, 15
         X = 8 - I
         DO 200 J = 1, 15
            Y = 8 - J
            IF(X**2 + Y**2.GE.50) GO TO 200
            K = K + 1
200      CONTINUE
100   CONTINUE
      WRITE(6, 10) K
10    FORMAT(1X, 'THE NUMBER OF POINTS IS', 2X, I5)
      STOP
      END
```

٥ – ١٢ أوجد كل الأرقام الأولية ذات الخانات الثلاث ، أى أوجد كل الأرقام الأولية ما بين 100 و 999 (قارن مع المثال ٥ – ٦).

لا يمكن أن يكون أحد هذه الأرقام الأولية زوجياً لذلك فهنا نعتبر الأعداد الصحيحة الفردية لـ K فقط بين 101 ، 999 لنرى ما إذا كانت K أولية . بالإضافة إلى ذلك ، إذا كانت K غير أولية ، فيجب أن يكون لـ K قاسم فردى M بين 3 و K/2 وبذلك ، نستخدم حلقة DO تكرارية بدليل M لتوليد القواسم المحتملة (الممكنة) لـ K المعطاة . تظهر خريطة سير العمليات وترجمتها إلى الفورتران فى شكل ٥ – ١٥ (أ) و (ب) ، على الترتيب .



شكل ٥ – ١٥ (أ)

```
C
C      PROGRAM PRINTING PRIME NUMBERS
C
       DO 200 K = 101, 999, 2
          KK = K/2
          DO 100 M = 3, KK, 2
             IF(K.EQ.(K/M)*M) GO TO 200
100       CONTINUE
          WRITE(6, 10) K
10        FORMAT(1X, I10)
200    CONTINUE
       STOP
       END
```

شكل ٥ – ١٥ (ب)

مســـائل تكميلية

حلقات DO التكرارية .

٥ — ١٤ حدد عدد مرات تنفيذ حلقة DO التكرارية ، وقيم الدليل التى تنفذ بها الحلقة التكرارية إذا كانت جملتها الأولى هى :

(١) DO 100 L = 2, 15, 3
(ب) DO 200 JOHN = 5. 11
(جـ) DO 300 I = 6, 9, 5
(د) DO 400 J = 8, 5, 4

٥ — ١٥ اكتشف الأخطاء ، إن وجدت فى كل جملة من جمل DO التكرارية .

(١) DO 700, LAMB = 1, 14, L,
(ب) DO 800 J = 7, M**2, 2
(جـ) DO 900 K = I, J, K,
(د) DO 1000 LONG = K234, K123, K345

٥ — ١٦ أوجد قيمة K بعد تنفيذ كل جزء من برامج الفورتران التالية :

(١)
```
      K = 3
      DO 100 J = 3, 7, 3
         K = K + J
100   CONTINUE
      K = 3*K
```

(ب)
```
      K = 3
      M = 2
      DO 200 J = 3, 7  M
         K = K + J
200   CONTINUE
      K = 3*K
```

(جـ)
```
      K = 3
      M = 2
      DO 300 J = 3, 7, M
         M = M + K
300   CONTINUE
      K = 3*K
```

(د)
```
      K = 3
      M = 2
      DO 400 J = M, 7, M
         K = J + K + M
         IF(K.GT.9) GO TO 10
400   CONTINUE
10    K = 3*K
```

٥ — ١٧ أوجد قيمة K النهائية بعد تنفيذ كل جزء من برامج الفورتران التالية :

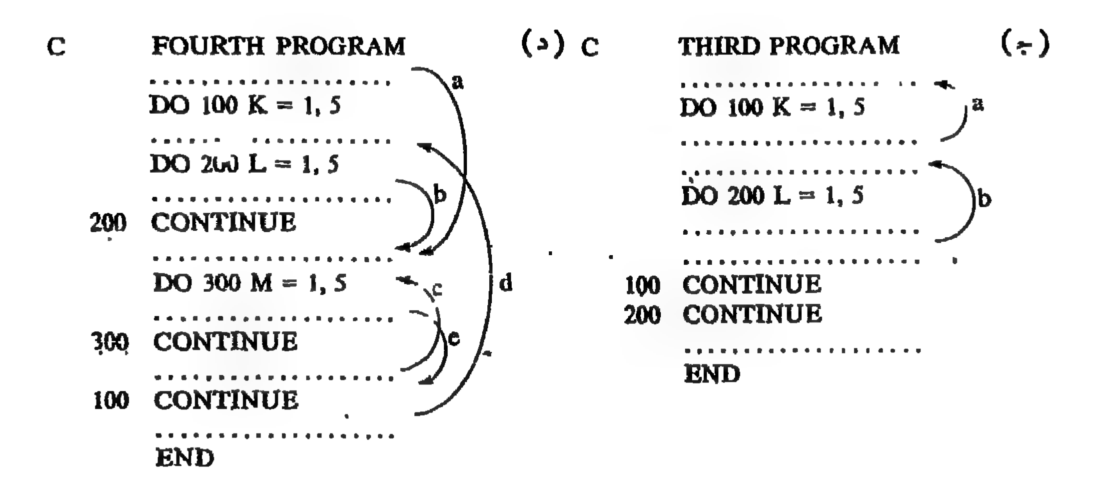
(١)
```
      K = 3
      M = 2
20    DO 10 J = 4, 9, M
         IF(J.EQ.6) GO TO 10
         K = K + 2*J
10    CONTINUE
      K = 3*K
```

(ب)
```
      K = 3
      M = 2
20    DO 10 J = 4, 9, M
         IF(J.EQ.6) GO TO 20
         K = K + J**2
10    CONTINUE
      K = 3*K
```

٥ — ١٨ افترض أن كل سهم مقوس يمثل انتقالا فى التحكم . أوجد الأخطاء فى كل برنامج من البرامج التالية :

(١)
```
C     FIRST PROGRAM
      ..................
      ..................   <-.
      DO 100 K = 1, 5        | a
      ..................   --'
      ..................
      ..................   <-.
100   CONTINUE               | b
      ..................   --'
      ..................
      END
```

(جـ)
```
C     SECOND PROGRAM
      ..................
      ..................   --.  a
      DO 100 K = 1, 5        |
      ..................     |
      ..................   <-+--.
      ..................   <-.  |
      DO 100 L = 1, 5        | b|
      ..................   --'  |
      ..................        |
      ..................   <----' 
100   CONTINUE               
      ..................      c
      ..................   --' 
      END
```



برامج

(كثير من المسائل هى نفسها كالمسائل الموجودة فى الفصل الرابع . بينما يجب أن تحل الآن باستخدام حلقة DO بدلا من استخدام عداد) .

٥ — ١٩ اكتب البرنامج الذى يطبع كل رقم فردى ذو خانتين N ومربعه N^2 وتكعيبه N^3 حيث تظهر قيم N المختلفة على أسطر مختلفة (قارن مع المسألة ٤ — ٩)

٥ — ٢٠ اكتب جزءاً من برنامج فورتران يطبع الرقم 20 عشرين مرة ، الرقم 19 تسع عشرة مرة ، الرقم 18 ثمانى عشرة مرة ، وهكذا .

٥ — ٢١ افرض عددين صحيحين موجبين N و K فى الذاكرة حيث K < N . اكتب جزء البرنامج مستخدما جملة DO حيث تطبع قيمة N ثم تطبع قيمة 1 — N ثم تطبع قيمة 2 — N وهكذا إلى أن نصل إلى K ونطبع قيمتها . (تلميح : انظر مثال ٥ — ٥ (ب)).

٥ — ٢٢ إذا أعطيت قيم A ، B ، N . اكتب جزء من برنامج فورتران ليحسب

$$\frac{1}{A}+\frac{1}{A+B}+\frac{1}{A+2B}+\frac{1}{A+3B}+\cdots+\frac{1}{A+NB}$$

٥ — ٢٣ اكتب البرنامج الذى يقرأ عدداً صحيحاً موجباً $N \geq 10$ ثم يحسب حاصل الضرب التالى إلى خمسة أماكن عشريه .

$$\frac{1}{1^2}\cdot\frac{3}{2^2}\cdot\frac{5}{3^2}\cdot\ldots\cdot\frac{2N-1}{N^2}$$

٥ — ٢٤ اكتب البرنامج الذى يطبع كل الأعداد الصحيحة الموجبة الفردية الأقل من 100 مع حذف تلك الأعداد الصحيحة التى تقبل القسمة على 7 :

1, 3, 5, 9, 11, 15, 17, 19, 23, . . . , 97, 99

٥ — ٢٥ افرض ان AMOUNT هى قيمة وديعة فائدتها RATE (مركبة سنوياً) ، والرقم N عدد السنوات كلها تقرأ كبيان . اكتب البرنامج الذى يطبع مقدار (AMOUNT) المبلغ كل سنة لمدة قدرها N من السنوات (قارن مع مسألة ٥ — ٨) .

٥ — ٢٦ افرض أننا أودعنا 500.00 $ كل سنة فى حساب توفير يعطى 7 فى المائة فائدة مركبة سنوياً . اكتب البرنامج الذى يطبع المبلغ الموجود فى الحساب كل سنة ولمدة 10 سنوات .

٥ — ٢٧ افرض أننا أودعنا $500.00 كل سنتين فى حساب توفير يعطى 7 فى المائة فائدة مركبة سنوياً . اكتب البرنامج الذى يطبع المبلغ الموجود فى الحساب كل سنة ولمدة 10 سنوات .

٥ — ٢٨ اقترض رجل مبلغ 300.00 $ من بنك بمعدل فائدة 1.5 فى المائة شهريا ، وهو يسدد $25.00 عند نهاية كل شهر . وبذلك ، فعند نهاية الشهر الأول يكون مدانا بالمبلغ التالى :

$$\begin{aligned} \text{AMOUNT} + \text{INTEREST} - \text{REPAYMENT} &= 300 + (1.5\%)(300) - 25 \\ &= 300 + 4.50 - 25 = 279.50 \end{aligned}$$

(ا) اكتب برنامج الفورتران الذى يطبع المبلغ المدان به كل شهر ولمدة سنة .

(ب) اكتب جزء من برنامج فورتران يحسب عدد الشهور التى يجب أن يدفع بها أقساط هذا القرض ، وقيمة آخر قسط .

٥ — ٢٩ أوجد عدد النقط ذات الاحداثيات الصحيحة التى تقع داخل القطع الناقص الممثل بالمعادلة $2x^2 + 3y^2 = 100$

٥ — ٣٠ ثلاثة أعداد صحيحة موجبة a و b و c حيث $a < b < c$ تشكل ثلاثية فيثاغورث إذا كانت $a^2 + b^2 = c^2$. على سبيل المثال 5 ، 4 ، 3 تشكل ثلاثية فيثاغورث حيث أن $5^2 = 4^2 + 3^2$. اكتب البرنامج الذى يجد كل ثلاثيات فيثاغورث a, b, c حيث $a, b < 25$.

٥ — ٣١ افرض أن كل بطاقة فى مجموعة بطاقات تحتوى على رقم حقيقى . أضيفت بطاقة مقدمة تحتوى على الرقم N وهو عدد البطاقات فى المجموعة . أوجد أصغر وأكبر قيمة .

٥ — ٣٢ افرض أنه تم تثقيب 25 عدد صحيح موجب على بطاقات ، عدداً فى كل بطاقة . اكتب البرنامج الذى يجد العدد التالى لأكبر عدد صحيح .

٥ — ٣٣ افرض مجموعة من البطاقات تحتوى على عدد واحد صحيح موجب لكل بطاقة . اكتب البرنامج الذى يجد أكبر عدد صحيح زوجى فى المجموعة ، أو يطبع NO EVEN INTEGER إذا كانت كل الأعداد الصحيحة فردية . اعتبر المجموعة لها بطاقة مقدمة تبين عدد البطاقات N فى المجموعة .

٥ — ٣٤ افرض مجموعة بطاقات تحتوى على عدد واحد صحيح فى كل بطاقة . اكتب البرنامج الذى يجد عدد الأرقام الصحيحة الزوجية ، وعدد الأرقام الصحيحة الفردية (اعتبر المجموعة لها بطاقة مقدمة) .

٥ — ٣٥ الثابت π يمكن أن يقرب بواسطة الصيغة الرياضية التالية :

$$\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots$$

اكتب البرنامج الذى

(ا) يجمع أول 1000 حد

(ب) يجمع أول 1000 حد فى الترتيب العكسى .

(سؤال : هل هناك أى فرق ؟)

٥ – ٣٦ اعتبر المعادلة $y = x^2 - 4x + 6$. اكتب برنامج الفورتران الذى يجد y لقيم x :

(ا) من 5– إلى 5 بمعامل زيادة 0.1

(ب) من I إلى J بمعامل زيادة $D = 1/N$ ، حيث I ، J ، N مثقبة على بطاقة و $I < J$

يجب أن تطبع كل قيمة لـ x وقيمة y المقابلة (لها على سطر مختلف)

٥ – ٣٧ اعتبر المعادلة $z = x^2 - 2xy + 3y^2 - 8x + 3y - 8$. اكتب البرنامج الذى يجد قيمة z لقيم x و y حيث كل من x و y تتراوح ما بين 3– و 3 بمعامل زيادة 0.2

٥ – ٣٨ عدل المسألة السابقة ٥ – ٣٧ بحيث يحدد البرنامج أيضاً . أقصى قيمة لـ z

٥ – ٣٩ يمكن أن نحسب قيمة تقريبية لـ جيب x بالتقدير الدائرى وذلك بتجميع أول N حد من المتسلسلة التالية :

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

(ا) اكتب البرنامج الذى يحسب الجيب لـ 3 درجات بالتقدير الدائرى باستخدام المتسلسلة السابقة و $N = 1, 2, \ldots\ldots, 8$

قارن كل إجابة بتلك التى نحصل عليها باستخدام الدالة المكتبية SINX

(ب) اكتب البرنامج الذى يقبل عدداً صحيحاً موجباً N وقيمة حقيقية x ويحسب قيمة جيب x باستعمال المتسلسلة السابقة .

(جـ) اكتب البرنامج الذى يقبل قيمة حقيقية x ويحسب قيمة x بتجميع حدود متتالية من المتسلسلة السابقة إلى أن تكون قيمة الحد المطلقة أقل من 10^{-15} .

اجابات للمسائل التكميلية المختارة

٥ – ١٤ (ا) خمس مرات 2 ، 5 ، 8 ، 11 ، 14 (ب) سبع مرات 5 ، 6 ، 7 ، 8 ، 9 ، 10 ، 11

(جـ) مرة واحدة 6 (د) مرة واحدة 8

٥ – ١٥ (ا) لا يجب أن تكون هناك فصلة بعد 700 أو بعد L

(ب) لا يمكن أن نستعمل M**2 كمامل

(جـ) لا يجب أن تكون هناك فصلة بعد K . متغير الدليل K لا يمكن أن يكون واحداً من المعاملات .

(د) لا توجد أخطاء .

٥ – ١٦ (ا) 36 (ب) 54 (جـ) خطأ لأن المعامل M تم تغيره (د) 39

٥ – ١٧ (ا) 81 (ب) الحاسب لن ينتهى من هذا الجزء من البرنامج (انظر المسألة المحلولة ٥ – ٤ (ب))

٥ – ١٨ (ا) النقل غير مسموح به .

(ب) تنقلات a ، c غير مسموح بهما .

(جـ) يجب أن تبدل جمل CONTINUE

(د) تنقلات a ، d غير مسموح بها .

الفصل السادس

المجموعات المتراصة والمتغيرات ذات الأدلة

٦ ــ ١ مقدمــــــة

إن المتغيرات التى استخدمناها إلى الآن تسمى أيضاً متغيرات بدون أدلة أو متغيرات غير متجهة ، وكل متغير من هذه المتغيرات غير المتجهة يمثل خلية ذاكرة يمكن تخزين قيمة واحدة بداخلها ، فمثلا ، يمكن اعتبار متغيرا باسم X كصندوق :

X	

أحياناً كثيرة يمكن أن نستخدم نفس الإسم للإشارة إلى قائمة من القيم لها نفس الخصائص المشتركة ، مثلا أرقام البطاقات الشخصية لطلبة فى فصل علم الحاسبات ودرجات الاختبار النهائى ، وهكذا ويمكن إنجاز ذلك باستخدام المجموعات المتراصة أو المتغيرات ذات الأدلة.

فى الرياضيات ، يستخدم رمز بأدلة مختلفة للاشارة إلى العناصر فى مجموعة معطاة على سبيل المثال S_1 ، S_2 ، ... S_{25} يمكن أن تشير إلى درجات 25 طالب فى فصل . وباستخدام الترميز بالأدلة يمكن حينئذ كتابة :

$$\sum_{i=1}^{25} s_i = s_1 + s_2 + \cdots + s_{25}$$

وذلك لتجميع الدرجات ، و

$$\left(\sum_{i=1}^{25} s_i\right)/25 .$$

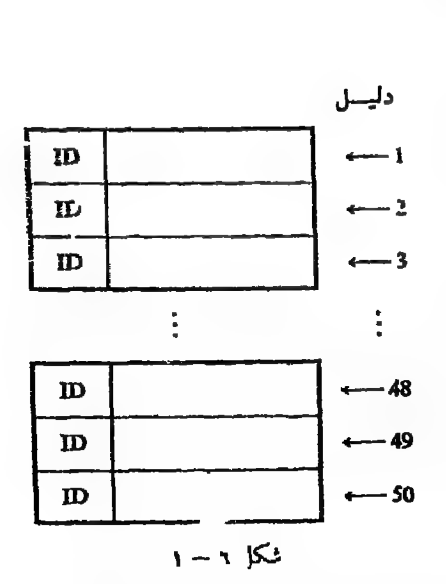
لإيجاد متوسط درجات الفصل ، بمعنى آخر فاستخدام الأدلة ضرورى جداً فى تطوير ترميز (مختصر) للأرقام المشمولة عند استخدام الجبر .

وفى الفورتران تستخدم الأدلة أيضاً ، ولكن لها مظهر مختلف . بمعنى أن كل جملة فورتران يجب أن تثقب (أو تكتب) على سطر واحد ، فتستخدم الأقواس لتحيط الأدلة ، فمثلا يمكن أن نكتب فى الفورتران S(1) ، S(2) ، ... ، S(25) بدلا من S_1 ، S_2 ، ... ، S_{25} . تلك المجموعات والتى تعرف عناصرها بدليل واحد تسمى فى الفورتران مجموعات متراصة خطية أو ذات بعد واحد ، المجموعات المتراصة المتشابهة التى تعرف عناصرها بدليلين أو أكثر تسمى مجموعات متراصة ذات أبعاد متعددة .

٦ ــ ٢ مجموعات متراصة ذات بعد واحد

المجموعات المتراصة الخطية أو ذات البعد الواحد ، والتى بها N عنصر هى قائمة أو سلسلة من عدد N من خلايا الذاكرة ، حيث يمكن أن تخزن قيمة واحدة بكل خلية . هذه الخلايا لها نفس الإسم ومجاورة بعضها لبعض . نتعرف على أى خلية محددة فى هذه المجموعة المتراصة ذات بعد الواحد بواسطة متغير صحيح K ، يرمز إلى مكانها فى المجموعة المتراصة . (وبذلك يجب أن تكون قيمة $1 \le K \le N$).

على سبيل المثال افرض أن ID هي مجموعة متراصة ذات بعد واحد بها 50 عنصر ، فيمكن أن نتصور ID صف من 50 صندوق كما فى الشكل ٦ – ١ . يشار إلى كل خلية فى المجموعة المتراصة ID باسمها ومكانها ، ويسمى دليل سفلى . بالتحديد تشير ID (1) إلى العنصر الأول من المجموعة المتراصة ، وتشير ID(2) إلى العنصر الثانى ، وهكذا . وعلاوة على ذلك إذا عرفت K وكانت $1 \leq K \leq 50$ فإن ID(K) تشير إلى العنصر رقم K من المجموعة المتراصة



شكل ٦ – ١

تطبق القواعد المستعملة فى تسمية المتغيرات العادية أيضاً على المجموعات المتراصة ، أى ، يتكون اسم المجموعة المتراصة من واحد إلى ستة حروف أبجدية رقمية يجب أن يكون الأول منها أبجدياً . إذا كان أيضاً اسم المجموعة المتراصة يبدأ بالحرف I أو J أو K أو L أو N فيمكن أن تحتوى المجموعة المتراصة على أعداد صحيحة فقط . بالمثل فالمجموعات المتراصة التى تبدأ بحروف أخرى يمكن أن تحتوى على أعداد حقيقية ، رغم أنه يمكن التغلب على التسمية التقليدية بجملة نوع (والتى تناقش فى قسم ٦ – ٢) ، ولكن يجب أن تكون عناصر المجموعة المتراصة دائماً من نفس النوع .

رغم أن أى عنصر من عناصر المجموعة المتراصة يحدد باسم المجموعة المتراصة متبوع بالدليل المناسب (محاط بأقواس) ، فيمكن أن يظهر كل عنصر فى المجموعة المتراصة فى التعبيرات الحسابية مثل المتغيرات غير المتجهة تماماً . على سبيل المثال التعبيرات التالية صحيحة لغوياً :

```
X = AMOUNT(5) + Q
PAY(K) = HOUR(K)*40.0
TAX(6) = 25.75
```

ومن ناحية أخرى ، لايمكن أن نستخدم أسماء المجموعات المتراصة نفسها فى الجمل الحسابية – على سبيل المثال ، إذا كانت GRADE هى مجموعة متراصة و SUM هى متغير عادى فالجمل التالية مثلا :

```
SUM = GRADE + SUM
GRADE = SUM + 10.0
GRADE = GRADE + 20.0
```

غير مسموح بها . وهى لأن الإسم GRADE يشير إلى المجموعة المتراصة بأكلها بينما الحسابات تشتمل على أماكن ذاكرة فردية .

يمكن أيضاً أن نشير إلى عناصر المجموعة المتراصة فى جمل الإدخال / الإخراج على سبيل المثال :

```
READ(5, 10) X, TAX(3)
```

تأمر الحاسب أن يقرأ رقمين . يخزن الرقم الأول فى المكان X ويخزن الثانى فى العنصر الثالث من المجموعة المتراصة المسماة TAX ونستطيع أن نطبع من الذاكرة أرقام تحقيق الشخصية ID(1) ، ID(2) ، . . . ID(K) كل رقم فى سطر باستخدام جزء البرنامج التالى :

```
      DO 100 I = 1, K
            WRITE(6, 20) ID(I)
 20         FORMAT(11X, I10)
100   CONTINUE
```

رغم أن أسماء المجموعات المتراصة لايمكن أن تظهر فى الجمل الحسابية ، لكنها يمكن أن تظهر فى جمل الإدخال / الإخراج وسيناقش هذا فيما بعد .

ملحوظة : يستخدم المصطلح متغيرات ذات أدلة فى كتب مختلفة ليعنى أشياء مختلفة . فبعض الكتب تستخدم متغيرات ذات أدلة لتعنى المجموعة المتراصة نفسها . فى حين أن كتباً أخرى تستخدم متغيراً ذا دليل ليعنى عنصراً فى المجموعة المتراصة وكتب أخرى ستقول أن العناصر ID(24) ، ID(3)... الخ ، هى متغيرات ذات أدلة. لاتستخدم كثير من كتيبات IBM فى لغة الفورتران المصطلح متغيراً ذا دليل على الإطلاق . ولكن كل الكتب تستخدم المصطلح متغيراً بدون دليل لتعنى نفس الشيء وهى المتغيرات العادية غير المتجهة .

٦ ـ ٣ جمل DIMENSION

قبل أن تستخدم أى مجموعة متراصة خطية أو متعددة الأبعاد فى برنامج . يجب أن تمد المترجم بالمعلومات الآتية :

١ ـ اسم المجموعة المتراصة

٢ ـ عدد الأدلة فى المجموعة المتراصة (أى ، ما إذا كانت المجموعة المتراصة ذات بعد واحد أو ذات بعدين ، الخ) .

٣ ـ العدد الكلى لأماكن الذاكرة التى ستخصص لهذا الإسم أو بتحديد أكثر القيمة العظمى لكل دليل .

يتم هذا باستخدام جملة النوع DIMENSION (وكما هو الحال فى كل جمل النوع ، فإن جملة DIMENSION جملة غير قابلة التنفيذ ويجب أن توضع قبل أى جملة قابلة التنفيذ فى البرنامج) .

وفيما يلى جملة DIMENSION النموذجية :

```
DIMENSION AMOUNT(100)
```

تدل هذه الجملة المترجم على أن AMOUNT هو اسم مجموعة متراصة خطية حقيقية يجب أن تخصص لها عدد من أماكن الذاكرة مقدارها 100

بدلا من ذكر المجموعات المتراصة فى جمل منفصلة فيمكن أن تعرف عدة مجموعات متراصة فى جملة DIMENSION واحدة. وعلى سبيل المثال :

```
DIMENSION ID(50), TAX(150)
```

تعرف ID على أنها مجموعة متراصة خطية صحيحة بها 50 عنصراً و TAX مجموعة متراصة خطية حقيقية بها 150 عنصراً . (شكل المجموعات المتراصة ذات الأبعاد المتعددة متشابهة وستناقش فيما بعد) . ترتيب ذكر المجموعات المتراصة فى جملة DIM ليس مهما ، طالما أن المجموعات المتراصة كلها متضمنة فى جمل DIMENSION ، لاحظ أيضاً أنه ليس هناك فصلة بعد كلمة DIMENSION. ولكن المجموعات المتراصة المختلفة تفصل فيما بينها بفصلات فى جملة DIMENSION .

عموماً الدليل الصفرى أو السالب غير مسموح به . ومن الواضح ، يجب ألا نستعمل دليلا أكبر من الحجم الأقصى المحدد فى جملة DIMENSION . على سبيل المثال ، إذا أعطينا جملة DIMENSION السابقة ، يجب ألا نستخدم ID(55) أو TAX(182) فى البرنامج . يجب توخى الحرص بصفة خاصة من هذه الناحية ، حيث أن مترجمات كثيرة لاتتأكد من صحة الأدلة . وبالتالى يمكن ألاتكتشف أخطاء من هذا النوع ولكنها ستنفذها مما يؤدى إلى نتائج مليئة بالأخطاء .

ذكرنا أن جملة DIMENSION تخصص العدد الأقصى من أماكن الذاكرة المطلوبة فى مجموعة متراصة ، رغم أن مخطط البرامج يمكن أن يستخدم جزءاً من خلايا المجموعة المتراصة فقط . عموماً هذا التخصيص الزائد قد يكون ضرورياً ، ولكن يجب أن يتم حسب المسألة نفسها ومتطلباتها . على سبيل المثال ، يمكن ألا يعرف العدد الحقيقى للطلبة الذين أدوا الاختبار مثلا ، بعض الطلبة قد يتغيبون) ، ولكن يمكن استخدام

```
DIMENSION TEST(60)
```

آخذين فى الاعتبار أن الحجم الطبيعى لفصل دراسى لن يسع أكثر من 60 . ونؤكد أن استعمال مجموعات متراصة كبيرة بدرجة غير معقولة سيكون مكلفاً بالنسبة إلى وقت التشغيل والبيئة الحسابية ككل .

ملاحظة : وأننا نؤكد أن جملة DIMENSION تخصص خلايا الذاكرة للمجموعات المتراصة أثناء عملية الترجمة للبرنامج ، وليس أثناء التنفيذ ، . من ثم فالدليل المذكور فى أى جملة DIMENSION يجب أن يكون ثابتاً صحيحاً بدون إشارة (الاستثناء الوحيد الوحيد قد يكون فى البرامج الفرعية SUBPROGRAMS أنظر الفصل السابع) وبالتالى فإن الجمل فى الشكل .

```
DIMENSION AMOUNT(K)
```

أو

```
N = 75
DIMENSION SCORE(N)
REAP(5, 10) I
DIMENSION GRADE(I)
```

غير مقبولة .

هناك نقطة أخيرة نريد أن نذكرها فى هذا القسم . إفرض أننا نريد SCORE كإسم لمجموعة متراصة خطية تحتوى على 50 عدداً صحيحاً على الأكثر (وليست أرقاماً حقيقية) يمكن عمل هذا باستعمال جمل النوع :

```
INTEGER SCORE
DIMENSION SCORE(50)
```

إلا أنه ، يمكن أن نستخدم جملة واحدة كبديل وهى :

```
INTEGER SCORE(50)
```

حيث أنها تعطى أيضاً المعلومات الهامة إلى المترجم . بالمثل :

```
REAL INVEST(150)
```

سعرف INVFST كاسم للمجموعة المتراصة الخطية التى بها 150 قيمة حقيقية . وكالمعتاد يمكن أن يظهر أكثر من بند من المعلومات فى جملة هذه ، ويجب فصل البنود عن بعضها بواسطة فصلات .

٦ ــ ٤ التعبيرات الرياضية للأدلة

إحدى المزايا الرئيسية لاستخدام المجموعات المتراصة هى أن الأدلة نفسها يمكن أن تكون متغيرات . فى الحقيقة يمكن أن تكون تعبيرات حسابية مقصورة على أحد الأشكال الآتية :

n, K, K + n, n*K, n*K ± m

حيث *n* ، *m* ثوابت صحيحة و K أى متغير صحيح . على سبيل المثال ، مايلى يعتبر صحيحاً لغوياً :

A(8), A(K), S(K − 2), A(3*K), A(4*K − 3)

ولكن التعبيرات الآتية لا تكون مقبولة عادة .

A(—4) (الدليل السالب غير مسموح به)

A(0) (الصفر غير مسموح به كدليل)

A(K*2) (يجب أن تكتب A(2*K))

A(K**2) (الدليل ليس أحد الأشكال السابقة)

A(.TRUE.) (لاتعتبر الأدلة قيماً منطقية (أنظر قسم ٩ ــ ٦))

A(3*M(2)) (متغيرات غير متجهة هى التى يمكن أن تظهر فقط فى تعبيرات الدليل)

اننا نذكر أن بعض الحاسبات الكبيرة الحديثة تسمح بأى تعبيرات حسابية صحيحة القيمة كأدلة ، ويجب أن تعدل القواعد السابقة تبعاً لذلك .

مع أننا إلى الآن قد ناقشنا فقط مجموعات متراصة ذات بعد واحد إلا أن القواعد السابقة على الأدلة تطبق أيضاً على المجموعات المتراصة متعددة الأبعاد .

٦ — ٥ امثلة لاستخدام المجموعات المتراصة

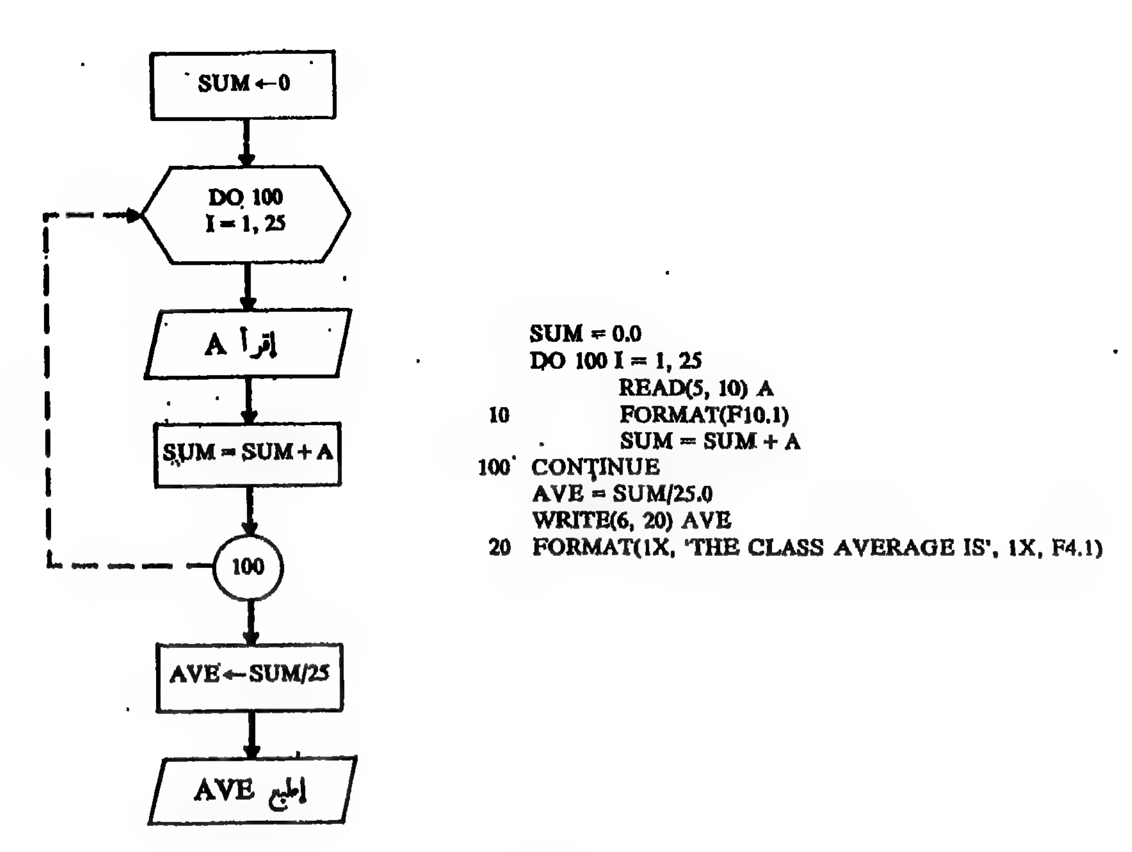
افرض أنه تم عقد امتحان لفصل دراسى أقل من 25 طالب ، وثقبت الدرجات على بطاقات ، درجة واحدة فى كل بطاقة . والمطلوب كتابة برنامج فورتران لعمل الآتى :

١ — إيجاد متوسط الفصل الدراسى

٢ — إيجاد عدد الطلبة الذين كانت درجاتهم أقل من المتوسط .

أولا سنناقش الحالة الأبسط وهى أن هناك فعلا 25 طالباً أدوا الامتحان (لإيجاد متوسط الفصل الدراسى ، يجب أن نجد أولا المجموع الكلى لكل الدرجات ويستلزم ذلك أن نتذكر المناقشة فى قسم ٤ — ١٠ لإيجاد مجموع عدة أرقام) .

طريقة وحيدة للخوض فى هذه المسألة هى باستخدام 25 إسماً متغيراً وليكن A_1 ، A_2 ، ... A_{25} وذلك لعدد 25 درجة . وهذا يؤدى إلى جمل طويلة جداً جداً تكون معرضة للخطأ . (انظر صفحة ١٤٤) بالإضافة إلى أننا سنحتاج 25 جملة IF حتى يقارن كل درجة بمتوسط الفصل . علاوة على ذلك ، يكون البرنامج غير قابل للسيطرة إذا تناولنا على سبيل المثال 100 رقم أو أكثر . لذلك يجب أن نبحث عن طريقة أخرى لحل المسألة .




```
      SUM = 0.0
      DO 100 I = 1, 25
         READ(5, 10) A
10       FORMAT(F10.1)
         SUM = SUM + A
100   CONTINUE
      AVE = SUM/25.0
      WRITE(6, 20) AVE
20    FORMAT(1X, 'THE CLASS AVERAGE IS', 1X, F4.1)
```

شكل ٦ — ٢

يمكن حساب متوسط الفصل الدراسى بطريقة مرتبة باستخدام خريطة سير العمليات وجزء البرنامج فى شكل ٦ – ٢ . (قارن مع شكل ٤ – ١٧) . ومع ذلك لاتوجد طريقة لمقارنة أى من الدرجات بمتوسط الفصل بماأن الدرجات لم تخزن فى الذاكرة . وبذلك لايمكن إيجاد عدد الطلبة الذين حصلوا على درجات أقل من المتوسط .

استخدام المتغيرات ذات الأدلة يعتبر ضرورياً للغاية إذا أردنا استخدام المخطط السابق ، ويكون فى استطاعتنا حساب عدد الطلبة الذين أخذوا درجات أقل من متوسط الفصل . بالتحديد ، تخزن الدرجات فى مجموعة متراصة خطية إسمها SCORE بها 25 عنصر . تظهر خريطة سير العمليات ومايقابلها بالفورتران فى شكل ٦ – ٣ .

ملاحظة : تخدم أول جملة WRITE فى شكل ٦–٣ غرضين ، فإلى جانب أنها تطبع الدرجات فهى تستخدم أيضاً كاختبار لاكتشاف ما إذا كانت المجموعة المتراصة SCORE قد قرئت بطريقة صحيحة أم لا. يسمى هذا الاختبار باختبار الصدى – ونلاحظ أن اختبار الصدى وسيلة مفيدة للغاية فى أحيان كثيرة وتستخدم فى اكتشاف الأخطاء ، أى تحديد أخطاء البرنامج . تضاف فى أحيان كثيرة جمل WRITE للبرامج المعقدة لتظهر نتائج وسيطة ، عندئذ ، يمكن تحديد وإيجاد الأخطاء بسهولة . وبعد اكتشاف كل الأخطاء (*bugs* أو errors) ، يمكن حينئذ إلغاء هذه الجمل . (رغم أن أول جمل READ و WRITE تظهر فى حلقة DO التكرارية الخاصة بهم ، فن الواضح أنه كان من الممكن إدماجهم مع حلقة DO الثانية وجعلها حلقة واحدة) .

ندرس الآن المسألة الأصلية للفصل الدراسى الذى لا يزيد عدد طلبته عن 25 طالباً . ولما كان المطلوب معرفة عدد الطلبة N الذين أدوا الامتحان لإيجاد متوسط الفصل ، فإنه يجب أن تعرف N فى البرنامج . نعتبر حالتين فى إحدى هاتين الحالتين تشمل مجموعة البيانات بطاقة مقدمة ، وفى الأخرى يضاف لمجموعة البيانات بطاقة خلفية (تذكر قسم ٤ – ١١ من أجل مناقشة بطاقات المقدمة والبطاقات الخلفية) .

حالة (أ)

إفرض أن مجموعة البيانات لها بطاقة مقدمة تحتوى على عدد الطلبة (N) المؤدين للامتحان . يمكن تعديل البرنامج فى شكل ٦ – ٣ بسهولة كما يلى : أضف الآتى بعد جملة DIMENSION

```
      READ(5, 40) N
40    FORMAT(I3)
      WRITE(6, 50) N
50    FORMAT('1', 'NUMBER TAKING EXAM IS', 1X, I3)
```

ثم بعد ذلك غير 25 بـ N فى جمل DO الثلاث ، وغير AVE = SUM/25.0 إلى :

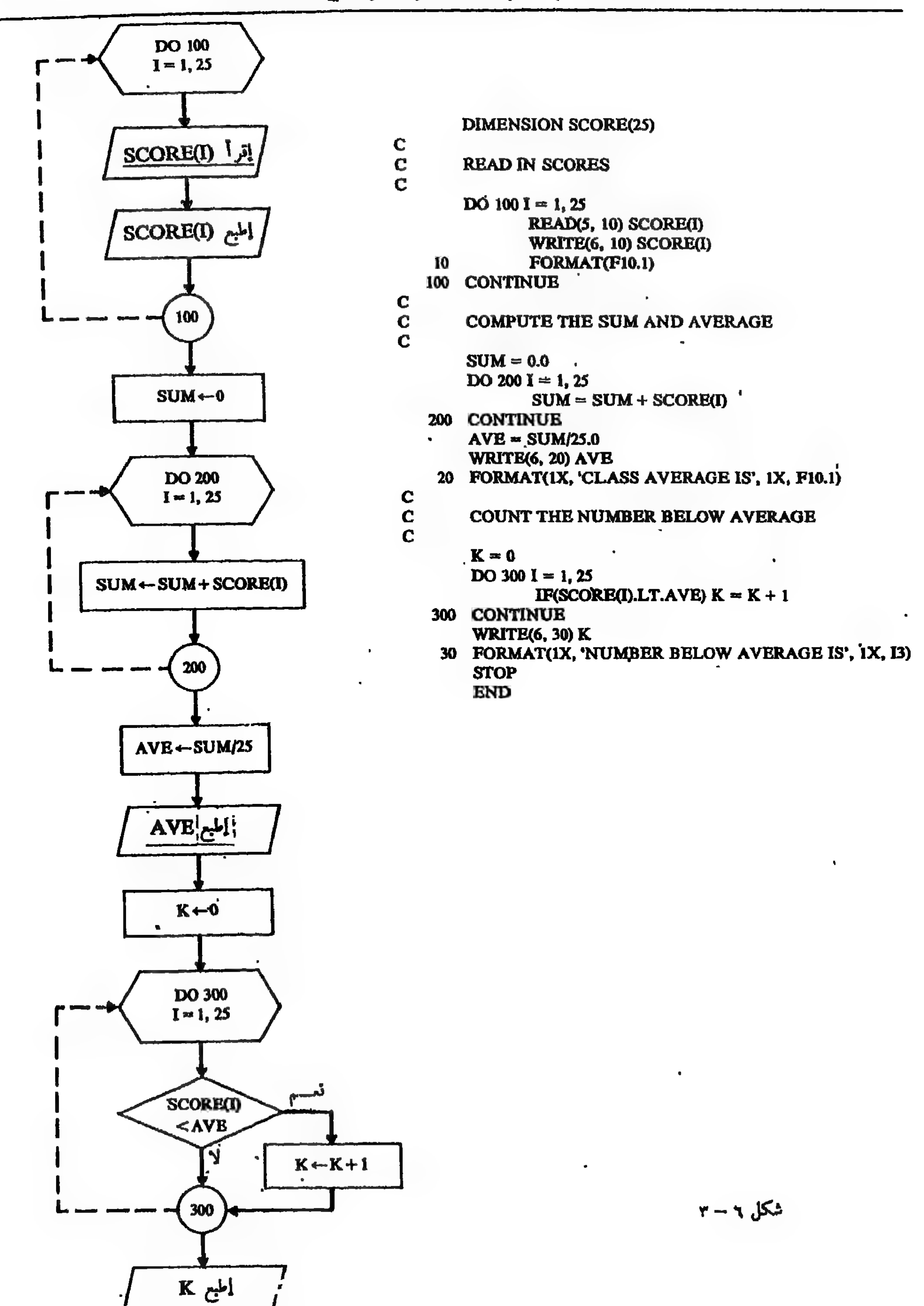
```
      AVE = SUM/FLOAT(N)
```

فيما عدا ذلك ، فالبرنامج مطابق للبرنامج السابق .

حالة (ب)

افرض أن مجموعة البيانات لها بطاقة خلفية تحتوى على رقم سالب . يخزن جزء برنامج الفورتران التالى الدرجات فى المجموعة المتراصة SCORE ويحسب أيضاً عدد الطلبة الذين أدوا الامتحان (N) :

```
      I = 0
111   READ(5, 60) A
 60   FORMAT(F10.1)
      IF(A.LT.0.0) GO TO 222
      I = I + 1
      SCORE(I) = A
      GO TO 111
222   N = I
```




```
      DIMENSION SCORE(25)
C
C     READ IN SCORES
C
      DO 100 I = 1, 25
         READ(5, 10) SCORE(I)
         WRITE(6, 10) SCORE(I)
10       FORMAT(F10.1)
100   CONTINUE
C
C     COMPUTE THE SUM AND AVERAGE
C
      SUM = 0.0
      DO 200 I = 1, 25
         SUM = SUM + SCORE(I)
200   CONTINUE
      AVE = SUM/25.0
      WRITE(6, 20) AVE
20    FORMAT(1X, 'CLASS AVERAGE IS', 1X, F10.1)
C
C     COUNT THE NUMBER BELOW AVERAGE
C
      K = 0
      DO 300 I = 1, 25
         IF(SCORE(I).LT.AVE) K = K + 1
300   CONTINUE
      WRITE(6, 30) K
30    FORMAT(1X, 'NUMBER BELOW AVERAGE IS', 1X, I3)
      STOP
      END
```

شكل ٦ – ٣

يمكن عمل هذا أيضاً باستخدام حلقة DO . نجعل الدليل يتراوح ما بين 1 إلى 26 حيث لايمكن أن يكون هناك أكثر من 25 درجة مع البطاقة الخلفية

```
      DO 500 I = 1, 26
            READ(5, 60) A
60          FORMAT(F10.1)
            IF(A.LT.0.0) GO TO 222
            SCORE(I) = A
500   CONTINUE
222   N = I - 1
```

لاحظ أن N = I — 1 حيث أن الدليل I يحسب أيضاً البطاقة الخلفية . حيث أن هناك 25 طالباً على الأكثر ، فسيكون هناك دائماً مخرج غير طبيعى من حلقة DO . وعلى ذلك تتحدد دائماً قيمة I فى الجملة 222 .

فيما يلى أمثلة أخرى تشتمل على مجموعات متراصة .

مثال ٦ — ١

افرض أن A(1)، A(2)، . . . ، A(N) مخزنة فى الذاكرة وأن N ≥ 2 . والمطلوب إيجاد أكبر الأعداد للمجموعة المتراصة A .

(أ) إحدى هذه الطرق هى إيجاد القيمة الحالية للرقم الأكبر أثناء المرور خلال المجموعة المتراصة (كما سبق أن ناقشنا ذلك فى م ٤ — ١٠) .

```
      XLAR = A(1)
      DO 100 I = 2, N
            IF(A(I).LE.XLAR) GO TO 100
            XLAR = A(I)
100   CONTINUE
```

(ب) الطريقة الثانية مفيدة عند فرز البيانات وفيها يتم تحديد مكان L للقيمة العظمى وبذا تكون A(L) هى القيمة العظمى . هذا النظام الحسابى (الخوارزم) مشابه للنظام السابق فيما عدا أننا الآن نتذكر المكان L للعدد الأكبر الحالى بدلا من العدد نفسه .

```
      L = 1
      DO 200 I = 2, N
            IF(A(I).LE.A(L)) GO TO 200
            L = I
200   CONTINUE
      XLAR = A(L)
```

فى الحالة التى لانعرف فيها أن N ≥ 2 (أى ، N يمكن أن تساوى 1) ، فإنه يجب تغيير جملة DO بحيث يتراوح الدليل I من 1 إلى N .

(هل يستطيع القارىء شرح ماهو الفرق ؟)

مثال ٦ — ٢

اعتبر أن A(1) ، A(2)... ، A(N) مخزنة فى الذاكرة . استعمل حلقة DO لطباعة A(N) ، A(N — 1) ، ....A(2) ، A(1) بحيث تكون كل واحدة فى سطر (أنظر مثال ٥ — ٥ (جـ))

كما سبق وأن ناقشنا من قبل فلا يمكن أن يكون معامل الزيادة لحلقة DO غير الميكلية سالباً ، ومن ثم فالبرنامج التالى غير صحيح :

```
      DO 300 I = N, 1, -1
         WRITE(6, 10) A(I)
10       FORMAT(10X, F10.1)
300   CONTINUE
```

وعلى أى حال لما كان الدليل I يتغير من 1 إلى N ، ونريد أن نطبع A(N) ، ... ، A(1) أى لدينا :

قيمة الدليل I	العنصر الذى سيطبع
1	A(N)
2	A(N − 1)
3	A(N − 2)
...	...
N	A(1)

هناك علاقة محددة بين الدليل I ودليل A فمجموعها دائماً N + 1 وبذلك سيطبع البرنامج التالى عناصر A فى الترتيب العكسى كما هو مطلوب :

```
      DO 300 I = 1, N
         K = N + 1 - I
         WRITE(6, 10) A(K)
10       FORMAT(10X, F10.1)
300   CONTINUE
```

٦ ـ ٦ المجموعات المتراصة ذات الأبعاد المتعددة

المجموعات المتراصة الخطية (ذات البعد الواحد) والتى تمت مناقشتها حتى الآن لها دليل واحد فقط ، ويسمح الفورتران أيضاً بمجموعات متراصة لها دليلان أو ثلاثة أدلة . تسمى هذه مجموعات متراصة ذات بعدين أو ثلاثة أبعاد ، على الترتيب . فى الحقيقة تسمح بعض الحاسبات الكبيرة بمجموعات متراصة لها سبعة أدلة . وفى هذا القسم ، سنناقش هذه المجموعات المتراصة ذات الأبعاد المتعددة .

تعرف المصفوفة رياضياً كمجموعة متراصة على شكل مستطيلا من الأعداد فمثلا ما يلى :

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

	رابع عمود	ثالث عمود	ثانى عمود	أول عمود
أول صف	a_{14}	a_{13}	a_{12}	a_{11}
ثانى صف	a_{24}	a_{23}	a_{22}	a_{21}
ثالث صف	a_{34}	a_{33}	a_{32}	a_{31}

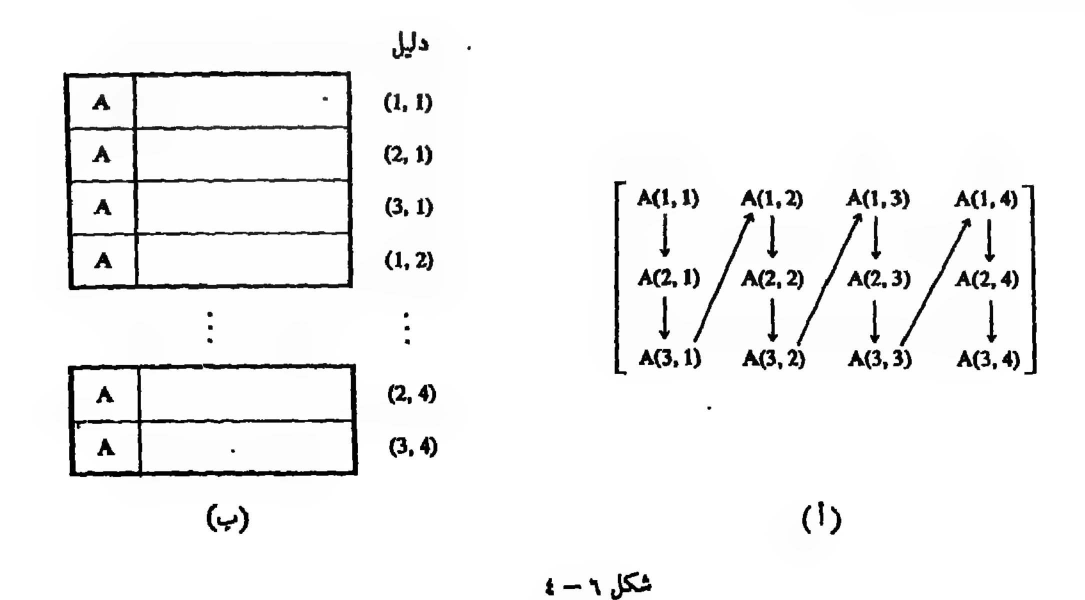
مصفوفة 3 فى 4 (وتكتب 4 × 3) أى ، مصفوفة بها ثلاثة صفوف (الأسطر الأفقية للأعداد) وأربعة أعمدة (الأسطر الرأسية للأعداد) نتعرف على كل عنصر فى المصفوفة بعددين صحيحين ، هما أدلته . يعرف الدليل الأول مكان العنصر فى الصف والدليل الثانى مكانه فى العمود . على سبيل المثال ، تظهر a_{21} فى الصف الثانى والعمود الأول ، وتظهر a_{23} فى الصف الثانى والعمود الثالث ، وهكذا . ونكتب عادة 3×4 بدلا من 3 فى 4 ("3 by 4")

وبطريقة مناظرة ، فى الفورتران المجموعة المتراصة ذات بعدين $(m \times n)$ هى قائمة بها عدد $m.n$ من خلايا الذاكرة (عناصر المجموعة المتراصة) حيث يمكن أن يتحدد أى عنصر معين بزوج من الأعداد الصحيحة K و L حيث $1 \le K \le m$ و $1 \le L \le n$. والأعداد الصحيحة K و L تسمى أدلة العنصر . الدليل الأول K يسمى صف العنصر والدليل الثانى L عمود العنصر . (مثل هذه المجموعات المتراصة ذات ـ بعدين ستسمى أيضاً مجموعات متراصة أو مصفوفة) .

على سبيل المثال ، افرض A هي اسم مجموعة متراصة ذات بعدين (4×3) . إذن A يكون بها $3 \times 4 = 12$ عنصراً .

$$A(1,1), A(1,2), A(1,3), A(1,4), A(2,1), A(2,2), A(2,3), \ldots, A(3,3), A(3,4)$$

لاحظ أن كل عنصر يحدد باسم المجموعة المتراصة يتبعه زوج من الأدلة . مفصولا ن بفصلة و محاطان بأقواس . مع أن A تصور عادة كمجموعة متراصة فى شكل مستطيل كما فى الشكل ٦ — ٤ (أ) ، إلا أنها ستمثل داخلياً فى الحاسب على أنها سلسلة من 12 خلية ذاكرة كما فى الشكل ٦ — ٤ (ب) . لاحظ أن العناصر تخزن عمودية ، حتى أن الدليل الأول (صف) يتغير أولا ثم يتغير الدليل الثانى (عمود) . وترتيب هذا التخزين موضح بالأسهم فى شكل ٦ — ٤ (أ)



شكل ٦ — ٤

بالمثل ، المجموعة المتراصة ذات الأبعاد الثلاثة $(m \times r \times s)$ هى قائمة بها عدد $m.r.s$ من خلايا الذاكرة (عناصر المجموعة المتراصة) وفيه يتحدد عنصر معين بثلاثة أعداد صحيحة J ، K ، L (الأدلة) ، حيث .

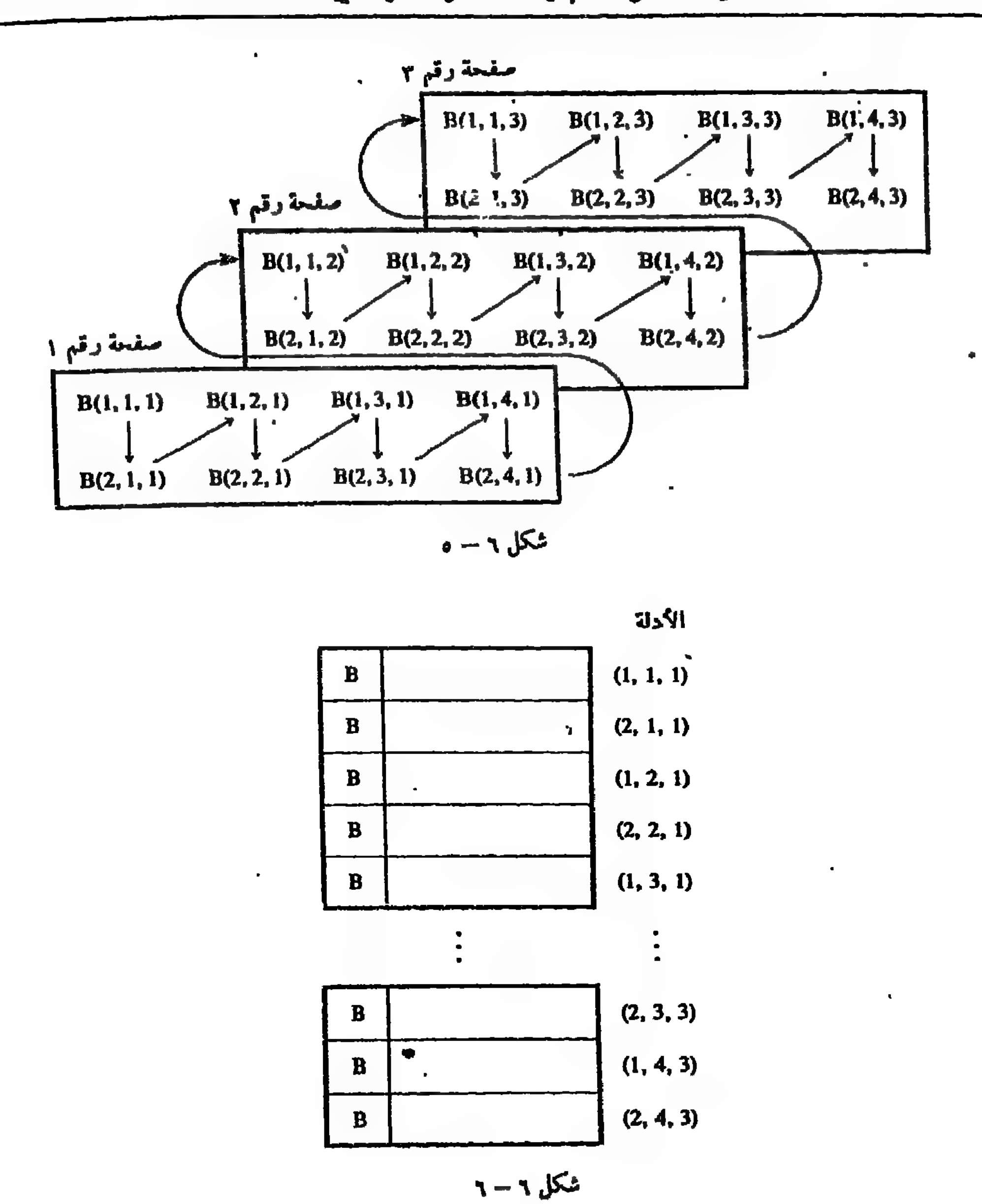
$$1 \le J \le m, \quad 1 \le K \le r, \quad 1 \le L \le s$$

على سبيل المثال ، افرض أن B مجموعة متراصة ذات ثلاثة أبعاد $(3 \times 4 \times 2)$ إذن B يكون بها $2 \times 4 \times 3 = 24$ عنصراً كما تظهر فى شكل ٦ — ٥ . لاحظ أن B قد صورت كأنها مكونة من ثلاث طبقات (صفحات) من مصفوفة 4×2 يسمى الدليل الأول الصف والدليل الثانى العمود ، والدليل الثالث صفحة العنصر ، على سبيل المثال $B(1, 4, 2)$ هى العنصر فى الصف الأول ، العمود الرابع والصفحة الثانية من B . ومع ذلك ، تمثل B داخلياً كسلسلة من 24 خلية ذاكرة مرصوصة بحيث يتغير الدليل الأول أولا (أكثر سرعة) ، والدليل الثانى ثانياً ، وهكذا ، كما فى الشكل ٦ — ٦. أى أن العناصر تنظم عمودياً من أول صفحة ثم عمودياً من ثانى صفحة ، ثم عمودياً من ثالث صفحة ويوضح ترتيب التخزين بالأسهم كما فى شكل ٦ — ٥ .

وعلى العموم فالمجموعة المتراصة ذات عدد p من الأبعاد ، أى $(n_1 \times n_2 \times \ldots \times n_p)$ هى قائمة بها $n_1.n_2.\ldots.n_p$ عنصراً من خلايا الذاكرة (عناصر المجموعة المتراصة) وفيه يتحدد عنصر معين بواسطة p من الأعداد الصحيحة K_1، K_2 ، ... K_p (الأدلة) حيث

$$1 \le K_1 \le n_1, 1 \le K_2 \le n_2, \ldots, 1 \le K_p \le n_p$$

يحدد عنصر معين باسم المجموعة المتراصة متبوعاً بأدلته p مفصولة عن بعضها بواسطة فصلات ومحاطة بأقواس .



شكل ٦ — ٥

شكل ٦ — ٦

المجموعات المتراصة ذات الأبعاد المتعددة يجب أن تعرف بنفس طريقة المجموعات المتراصة ذات بعد واحد على سبيل المثال :

DIMENSION ID(25), MONEY(3, 2, 7), TAX(6, 5)

ستبلغ المترجم أن :

١ — ID مجموعة متراصة ذات بعد واحد يحتمل أن يكون له 25 عنصراً (أي ، سيخصص له 25 خلية ذاكرة مع عدم الالتزام باستخدام كل الخلايا بالبرنامج) .

٢ — MONEY مجموعة متراصة ذات ثلاثة أبعاد ويحتمل أن يكون بها 42 = 7 × 2 × 3 عنصراً .

٣ — TAX مجموعة متراصة ذات بعد ين ويحتمل أن يكون لها 30 = 5 × 6 عنصراً .

اننا نؤكد أن اسماء المجموعات المتراصة متعددة الأبعاد والأشكال لعناصرها تتبع نفس قوانين المجموعات المتراصة ذات البعد الواحد .

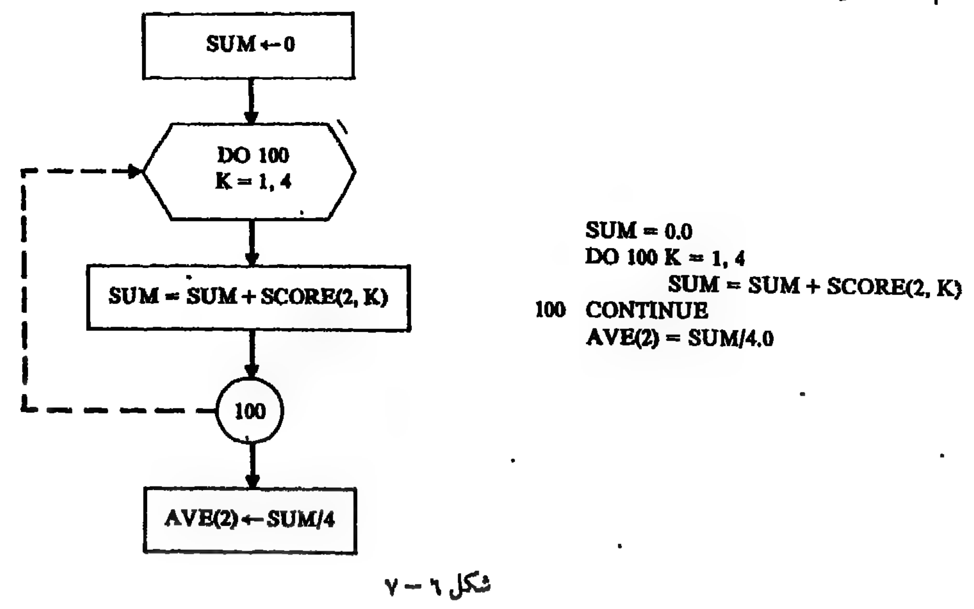
مثال ٦ — ٣

إفرض أنك أعطيت 4 اختبارات لفصل به 25 طالباً . يمكن أن تنظم درجات الاختبارات هذه في جدول كما يلي :

طالب	اختبار رقم ١	اختبار رقم ٢	اختبار رقم ٣	اختبار رقم ٤
1	78	83	80	87
2	100	100	92	100
⋮	⋮	⋮	⋮	⋮

إفرض أن درجات الاختبار هذه تم تخزينها في مجموعة متراصة حقيقية في شكل مصفوفة SCORE أبعادها (4 × 25) . وبذلك SCORE (I, K) تعطى درجة الطالب I في الاختبار *K* . بالتحديد ، فالصف الثاني :

يعطى درجات الاختبارات الأربعة للطالب الثاني . لترجمة متوسط الطالب ، سنجعل AVE(I) تشير إلى متوسط اختبار الطالب I يمكن أن نستخدم خريطة سير العمليات في شكل ٦ — ٧ ومايقابلها بالفورتران لإيجاد AVE(2) ، أي متوسط درجات الطالب الثاني .



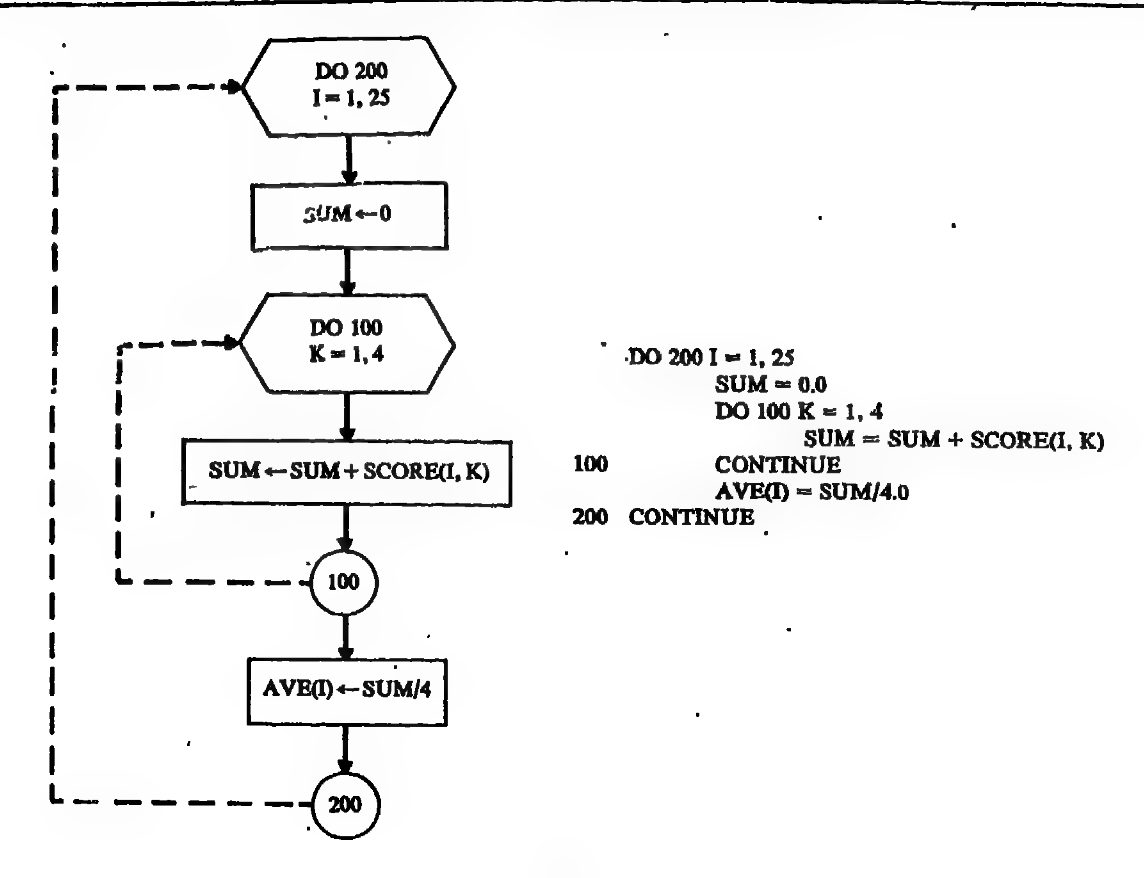

```
      SUM = 0.0
      DO 100 K = 1, 4
         SUM = SUM + SCORE(2, K)
100   CONTINUE
      AVE(2) = SUM/4.0
```

شكل ٦ — ٧

وبصورة عامة يمكن أن نجد AVE(I) متوسط درجات الطالب I بجزء من برنامج الفورتران التالي :

```
      SUM = 0.0
      DO 100 K = 1, 4
         SUM = SUM + SCORE(I, K)
100   CONTINUE
      AVE(I) = SUM/4.0
```

ومن ثم لحساب متوسط كل طالب من الـ25 طالباً . سنحتاج إلى حلقة DO متداخلة حيث حلقة DO الخارجية تنفذ 25 طالباً . خريطة سير العمليات تظهر في الشكل ٦ — ٨ وكذلك مايقابلها بالفورتران .



شكل ٦ – ٨

٦ ــ ٧ ادخال / اخراج مجموعة متراصة وحلقات DO الضمنية

افرض A و B مجموعات متراصة خطيةمكونة من أربعة وثلاثة عناصر ، على الترتيب . وكما ذكرنا سابقا ، يمكن أن تظهر عناصر A و Bفى جمل I/O (إدخال/ إخراج) كمتغيرات عادية . على سبيل المثال :

```
READ(5, 10) A(3), B(2), J, X
```

تخبر الحاسب أن يقرأ قيم لـ A(3) و B(2) وكذلك J وX . بالتحديد يمكن أن ندخل المجموعات المتراصة A و B بأكلها بذكر كل عناصرها مفردة ، على سبيل المثال :

```
READ(5, 20) A(1), A(2), A(3), A(4), B(1), B(2), B(3)
```

واضح أن هذا الأسلوب الفنى لن يكون مناسباً إذا كان هناك مئات من العناصر فى المجموعات المتراصة . سنناقش الآن طرقاً أخرى لإدخال / إخراج المجموعات المتراصة .

(أ) أسماء المجموعات المتراصة فى جمل (I/O) (الإدخال / الإخراج) :

يمكن أن نقرأ أو نطبع مجموعة متراصة بأكلها باستخدام إسم المجموعة المتراصة وذلك فى جملة الإدخال / الإخراج فقط . على سبيل المثال ، يمكن أن نعيد كتابة جملة READ السابقة بطريقة مكافئة .

```
READ(5, 20) A, B
```

ومع ذلك ، يجب أن نفهم بوضوح ما يترتب على أمر I/O مثل هذا :

١ – أولا سيخزن أو سيطبع المجموعة المتراصة بأكملها أى أن كل أماكن الذاكرة المخصصة بجملة DIMENSION ستأخذ قيمة . وتبعاً لذلك ، إذا عرفت SCORE كمجموعة متراصة كما يلى :

```
      DIMENSION SCORE(35)
```

إذن يجب ألا نستخدم :

```
      READ(5, 30) SCORE
30    FORMAT(5F10.1)
```

إذا أردنا أن نقرأ 25 قيمة فقط داخل SCORE .

٢ – ستخصص أو ستطبع قيم لعناصر المجموعة المتراصة تبعاً للترتيب تخزين المجموعة المتراصة (شكل ٦ – ١ ، ٦ – ٤ ، ٦ – ٦) ، حيث يتغير الدليل الأول بسرعة أكبر ، والدليل الثانى بعده ، وهكذا .

مثال ٦ – ٤

إفرض E و D و C مجموعات متراصة معرفة بالجملة :

```
      DIMENSION C(3), D(2, 3, 2), E(3, 2)
```

إذن فزوج الجمل WRITE–FORMAT

```
      WRITE(6, 30) C, D, E
30    FORMAT(6X, 4F10.2)
```

تأمر الحاسب أن يطبع مجموع القيم التالية :

$$3 + 2\cdot3\cdot2 + 3\cdot2 = 21$$

أى 3 من المجموعة المتراصة C و12 من المجموعة المتراصة D و6 من المجموعة المتراصة E . حيث أن جملة FORMAT لها أربع مواصفات حقول رقمية فقط ، ستطبع أربعة أرقام على كل سطر ، وستستعمل جملة FORMAT مرة أخرى وأخرى إلى أن تتم طباعة كل القيم . وبذلك سيبدو الخرج كما فى الشكل ٦ – ٩ .

C(1)	C(2)	C(3)	D(1, 1, 1)
D(2, 1, 1)	D(1, 2, 1)	D(2, 2, 1)	D(1, 3, 1)
D(2, 3, 1)	D(1, 1, 2)	D(2, 1, 2)	D(1, 2, 2)
D(2, 2, 2)	D(1, 3, 2)	D(2, 3, 2)	E(1, 1)
E(2, 1)	E(3, 1)	E(1, 2)	E(2, 2)
E(3, 2)			

شكل ٦ – ٩

تذكر أنه إذا استعملت جملة FORMAT مختلفة فى المثال السابق وليكن :

```
30    FORMAT(6X, 6F10.2)
```

فسوف يشمل الخرج ستة أرقام على كل سطر . ومع ذلك لن يتغير ترتيب الأرقام بجمل FORMAT المختلفة .

(ب) حلقات DO الضمنية

إفرض أننا نريد أن نخزن أو نطبع جزءاً من مجموعة متراصة أو افرض أننا لانريد أن نخزن أو نطبع مجموعة متراصة بترتيبها الموصوف فى التخزين . قد يمكننا عمل هذا باستخدام حلقة DO . على سبيل المثال ، إفرض أن AMOUNT مجموعة متراصة خطية محتمل أن تحتوى على 100 عنصر . إذن فحلقة DO التالية :

```
      DO 100 I = 11, N, 1
             READ(5, 10) AMOUNT(I)
100   CONTINUE
```

تأمر الحاسب أن يقرأ قيم AMOUNT(11) ، AMOUNT(12) حتى ومتضمناً AMOUNT(N) ، حيث N تم تعريفها مسبقاً و $N \leq 100$. يمكن أن ننجز نفس الشيء باستخدام جملة READ التالية :

```
READ(5, 20) (AMOUNT(I), I = 11, N, 1)
```

التعبير

```
(AMOUNT(I), I = 11, N, 1)
```

يسمى حلقة DO الضمنية حيث أنها تسبب حدوث حلقة تكرارية بداخل جملة READ والرمز :

```
I = 11, N, 1
```

له نفس المعنى هنا كما هو فى حلقة DO ، أى ، 11 هى القيمة الابتدائية و N هى القيمة النهائية ، و 1 هو مقدار الزيادة .

تذكر أيضا أن حلقة DO الضمنية أكثر تغيراً من حلقة DO السابقة . بالتحديد كل مرة تصادف جملة READ فى حلقة DO السابقة يجب أن تقرأ بطاقة بيانات أخرى بغض النظر عن كيفية تحديد جملة FORAMT المرقة 10 . وتبعاً لذلك يجب وضع قيم المدخلات كل على بطاقة منفصلة . من ناحية أخرى ، فإننا نمر ، فى حلقة DO الضمنية ، على جملة READ واحدة فقط ، وبذلك يمكن وضع المدخلات على بطاقة أو أكثر معتمداً فى ذلك على جملة FORMAT .

فيما يلى الشكل العام لحلقة DO الضمنية لجمل I/O .

```
(VN(I), I = IN, IE, IC)
```

حيث :

VN اسم المجموعة المتراصة

I الدليـــل

IN القيمة الابتـــدائية لـ I

IE قيمة الاختبار أو النهاية لـ I

IC معامل الزيـــادة

تذكر أن كلا من IN ، IE و IC يمكن أن يكون عددا صحيحاً موجباً أو متغيراً صحيحاً . أيضاً ، إذا كانت قيمة IC هى 1 فإنه يمكن حذفها . وهنا نؤكد أن الأقواس الخارجية والفصلة قبل الدليل ضرورية للغاية عند استخدام حلقة DO الضمنية .

يمكن استخدام الدليل I السيطرة على أكثر من مجموعة متراصة واحدة . على سبيل المثال :

```
READ(5, 30) (A(K), B(3, K), C(K, K, 2), K = 2, 9, 3)
```

تخبر الحاسب أن يدور خلال قائمة المتغيرات أولا عندما K = 2 . ثم عندما K = 5 ، وأخيراً عندما K = 8 ، من ثم يكون جملة READ مكافئة لـ :

```
      READ(5, 30) A(2), B(3, 2), C(2, 2, 2), A(5), B(3, 5), C(5, 5, 2),
     1            A(8), B(3, 8), C(8, 8, 2)
```

عموماً ليس من الضرورى أن تكون المتغيرات فى حلقة DO الضمنية ذات دليل أو تتضمن العداد .

مثال ٦ — ٥

(أ) إفرض X مجموعة متراصة خطية بها 100 عنصر مخزنة فعلا فى الذاكرة يمكن أن تطبع المجموعة المتراصة كل عنصر منها على سطر بواسطة :

```
      WRITE(5, 20) (X(I), I = 1, 100)
   20 FORMAT(2X, F8.2)
```

أو

```
      DO 100 I = 1, 100
         WRITE(5, 10) X(I)
   10    FORMAT(2X, F8.2)
  100 CONTINUE
```

يمكن أن تطبع المجموعة المتراصة كل أربعة عناصر منها على سطر بواسطة :

```
      DO 100 I = 1, 100, 4
         WRITE(5, 10) X(I), X(I + 1), X(I + 2), X(I + 3)
   10    FORMAT(4(2X, F8.2))
  100 CONTINUE

      WRITE(5, 20) (X(I), I = 1, 100)
   20 FORMAT(4(2X, F8.2))
```

وبذلك نرى سهولة تغيير حلقة DO الضمنية عن حلقة DO لعمليات I/O .

(ب) اعتبر زوج الجمل WRITE-FORMAT التالى :

```
      WRITE(6, 40) (A(I), B, X(3), Y(I, I), I = 1, 8, 3)
   40 FORMAT(6(2X, F8.2))
```

تنفذ حلقة DO داخل جملة WRITE ثلاث مرات ، أولا مع I = 1 ثم I = 4 وأخيراً مع I = 7 وعلى ذلك تكون جملة WRITE مكافئة لما يلى :

```
      WRITE(6, 40) A(1), B, X(3), Y(1, 1), A(4), B, X(3),
     1             Y(4, 4), A(7), B, X(3), Y(7, 7)
```

لاحظ أن B و X(3) يظهران ثلاث مرات . ستسبب جملة FORMAT طبع ستة أرقام فى السطر ، بحيث يبدو الخرج كالتالى :

```
A(1)   B        X(3)   Y(1, 1)   A(4)   B
X(3)   Y(4, 4)  A(7)   B         X(3)   Y(7, 7)
```

(حـ) سيطبع جزء البرنامج التالى:

```
      WRITE(6, 50) (I, X(I), I = 1, 3)
   50 FORMAT(1X, 'A(', I1, ') =', F10.2)
```

ما يلى :

```
A(1) = XXXXXXX.XX
A(2) = XXXXXXX.XX
A(3) = XXXXXXX.XX
```

(ح) حلقات DO الضمنية المتداخلة

حلقة DO الضمنية الموصوفة أعلاه يمكن أن تعمم لتسيطر على مجموعات متراصة متعددة الأبعاد (هذا التعميم مشابه لحلقات DO المتداخلة) . إفرض ، على سبيل المثال ، أن F قد عرفت على أنها مجموعة متراصة (2 × 3) والبرنامج يحتوى على الجملة

```
READ(5, 10) ((F(I, J), J = 1, 2), I = 1, 3)
```

فالدليل الخارجى I مشابه لدليل حلقة DO الخارجية ، والدليل الداخلى J مشابه لدليل حلقة DO الداخلية . أى ، تنفذ حلقة DO الضمنية الداخلية بالكامل لكل قيمة من دليل حلقة DO الضمنية الخارجية . وبذلك تكون جملة READ مكافئة لـ :

```
READ(5, 10) (F(1, J), J = 1, 2), (F(2, J), J = 1, 2), (F(3, J), J = 1, 2)
```

التى هى مكافئة لـ :

```
READ(5, 10) F(1, 1), F(1, 2), F(2, 1), F(2, 2), F(3, 1), F(3, 2)
```

لاحظ أن جملة READ العليا تقرأ القيـم إلى داخل F صفاً بصف . من ناحية أخرى الجملة :

```
READ(5, 20) ((F(I, J), I = 1, 3), J = 1, 2)
```

(لاحظ أنه تم تبديل الأدلة I ، J) تكافىء الجملة

```
READ(5, 20) F(1, 1), F(2, 1), F(3, 1), F(1, 2), F(2, 2), F(3, 2)
```

وفى هذه الحالة ، تقرأ القيـم إلى داخل F عموداً بعمود .

يمكن أن يكون لحلقات DO المتداخلة تركيبات أكثر تعقيداً ويمكن أن تظهر أيضاً فى جمل WRITE بنفس الطريقة .

مثال ٦ - ٦

(أ) إدرس الجملة

```
WRITE(6, 11) ((B(L, M), L = 2, 10, 2), M = 1, 11, 3)
```

أولا ، تنفذ حلقة DO الداخلية (B(L, M), L = 2, 10, 2) عندما يكون M = 1 ، M = 4 ، ثم M = 7 ، وأخيراً M = 10 هكذا تكون الجملة مكافئة لـ :

```
      WRITE(6, 11) (B(L, 1), L = 2, 10, 2), (B(L, 4), L = 2, 10, 2),
     1             (B(L, 7), L = 2, 10, 2), (B(L, 10), L = 2, 10, 2)
```

وهى مكافئة لـ :

```
      WRITE(6, 11) B(2, 1), B(4, 1), B(6, 1), B(8, 1), B(10, 1), B(2, 4), B(4, 4),
     1             B(6, 4), B(8, 4), B(10, 4), B(2, 7), B(4, 7), B(6, 7), B(8, 7),
     1             B(10, 7), B(2, 10), B(4, 10), B(6, 10), B(8, 10), B(10, 10)
```

لاحظ أن الدليل الثانى M فى B (L, M) يظهر فى حلقة DO الخارجية ، بينما يظهر الدليل الأول I فى F (I, J) فى حلقة DO الخارجية فى المثال السابق .

(ب) إفرض B مجموعة متراصة (3 × 5 × 4) وأفرض أننا نريد أن نطبع قيم B من الصفحة الأولى صفا بصف ثم من الصفحة الثانية صفاً بصف وأخيراً من الصفحة الأخيرة صفاً بصف . يمكن أن يتم هذا بزوج الجمل WRITE-FORMAT :

```
      WRITE(6, 50) (((B(K1, K2, K3), K2 = 1, 5), K1 = 1, 4), K3 = 1, 3)
   50 FORMAT(5(2X, F8.2))
```

فى الحقيقة ، ستسمح جملة FORMAT بطباعة خمسة أرقام فقط على السطر ، بحيث تظهر الصفوف المختلفة لـ B على أسطر مختلفة من صفحة الطباعة .

٦ ــ ٨ مثال لأسلوب البرمجة الفنى

يمكن أن يكون هناك عدة أنظمة حسابية (خوارزميات) لحل وصياغة أى مسألة ويكون البعض أفضل من البعض الآخر ، أى ، يحتاج البعض عدد خلايا ذاكرة أقل و / أو عمليات أقل . وتكرس الفصل الثامن لعدة أساليب للبرمجة الفنية . وندرس هنا مسألة بسيطة توضح النقط الدقيقة الماكرة التى يمكن أن تبرز أثناء الحل .

إفرض أننا نريد أن نحرك العناصر A(1) ، A(2) ، ...A(N) فى مجموعة متراصة ما أسفل القائمة مكاناً واحداً :

A(1) A(2) A(3) ... A(N − 1) A(N)

أى ، حتى تحتوى A(2) قيمة A(1) القديمة ، وتحتوى A(3) قيمة A(2) القديمة وهكذا . فالشىء البديهى أن نبدأ بتخصيص A(1) إلى A(2) ثم A(2) إلى A(3) وهكذا إلى أن تخصص A(N) إلى A(N + 1) . يمكن عمل هذا بحلقة DO التكرارية :

```
      DO 100 I = 1, N
         A(I + 1) = A(I)
100   CONTINUE
```

ومع ذلك لن يتم تنفيذ هذا الإجراء . السبب هو أنه عند تخصيص A(1) إلى A(2) فى بادىء الأمر تم إزالة القيمة الأصلية لـ A(2) وبذلك فإن تخصيص A(2) التالى إلى A(3) سيخزن القيمة الأصلية لـ A(1) فى A (3) – ومن ثم فبواسطة الإجراء السابق فإن القيمة الأصلية لـ A(1) ستخصص إلى كل أماكن التخزين A(2) ، A(3) ، ...A(N + 1) .

والطريقة لتصحيح الإجراء السابق هى أن تخزن القيم الأصلية لـ A مؤقتاً فى مراكز تخزين أخرى . يمكن عمل هذا باستخدام مجموعة متراصة أخرى B . بالتحديد ، نستخدم حلقة DO لتخصيص A(I) إلى B(I + 1) بعد ذلك نستخدم حلقة DO أخرى لنقل العناصر إلى A مرة أخرى بتخصيص B(I + 1) إلى A(I + 1) . وذلك يعنى أن نستخدم ضعف عدد خلايا الذاكرة التى فى A لمجرد نقل عناصر A إلى أسفل مكان واحد ، أى إذا احتلت A 1000 خلية ذاكرة إذن سنحتاج 1000 خلية ذاكرة أخرى لـ B .

طريقة أخرى ، نحتاج فيها إلى خليتين إضافيتين من خلايا الذاكرة فقط ، تخزن أولا A(2) فى مكان مؤقت TEMPA قبل تخصيص A(1) إلى A(2) ثم تخزن A(3) فى مكان مؤقت TEMPB قبل تخصيص TEMPA إلى A(3)

```
TEMPA ← A(2)
A(2)  ← A(1)
TEMPB ← A(3)
A(3)  ← TEMPA
```

لاحظ أن A(3) تحتوى الآن A(2) الأصلية . ثم نستعمل بعد ذلك TEMPA مرة أخرى لتخزين A(4) قبل تخصيص TEMPB إلى A(4) وهكذا . يمكن إنجاز ماسبق بحلقة DO التكرارية . رغم أننا استخدمنا خليتين إضافيتين للذاكرة فقط إلا أن النظام الحسابى (الخوارزم) يحتاج فعلا إلى 2N — 1 تخصيصاً .

فى الحقيقة أن أفضل طريقة لا تحتاج أى خلية ذاكرة إضافية وتحتاج عدد N تخصيصات فقط . وهى أننا ، نحرك ببساطة A(N) إلى A(N+1) ثم A(N — 1) إلى A(N) . وهكذا أى نعطى سلسلة الأوامر التالية :

```
A(N + 1) ← A(N)
A(N)     ← A(N − 1)
    ⋮
A(2)     ← A(1)
```

فيما يلى الفورتران المكافىء :

```
      DO 100 J = 1, N
         K = N + 1 − J
         A(K + 1) = A(K)
100   CONTINUE
```

أو ببساطة

```
      DO 100 J = 1, N
         A(N − J + 2) = A(N − J + 1)
100   CONTINUE
```

(أنظر المثال ٦ ــ ٢ لترى السبب فى استخدام المتغير الجديد K)

مســـائل محلولة

مجموعات متراصة

٦ – ١ حدد عدد العناصر فى المجموعات المتراصة تبعاً لجملة DIMENSION التالية :

```
DIMENSION A(25), B(3, 8, 4), K(7, 5)
```

A بها 25 عنصراً .

B بها $3 \times 8 \times 4 = 96$ عنصراً .

C بها $7 \times 5 = 35$ عنصراً .

٦ – ٢ أوجد عدد العناصر فى المجموعة المتراصة X المعرفة بالآتى :

```
DIMENSION X(2, 4, 3)
```

صف الخرج إذا نفذنا الجملتين الآتيتين :

```
   WRITE(6, 10) X
10 FORMAT(6(2X, F12.2))
```

X بها $2 \times 4 \times 3 = 24$ عنصراً . ستطبع كما هى مخزنة ، اى ، بحيث يتغير الدليل الأول (صف) بأقصى سرعة ، يليه الدليل الثانى (عمود) وأخيراً الدليل الثالث (صفحة) . أيضاً ، سيكون هناك ستة أرقام فى السطر . وبذلك يظهر الخرج كالتالى :

```
X(1, 1, 1)   X(2, 1, 1)   X(1, 2, 1)   X(2, 2, 1)   X(1, 3, 1)   X(2, 3, 1)
X(1, 4, 1)   X(2, 4, 1)   X(1, 1, 2)   X(2, 1, 2)   X(1, 2, 2)   X(2, 2, 2)
X(1, 3, 2)   X(2, 3, 2)   X(1, 4, 2)   X(2, 4, 2)   X(1, 1, 3)   X(2, 1, 3)
X(1, 2, 3)   X(2, 2, 3)   X(1, 3, 3)   X(2, 3, 3)   X(1, 4, 3)   X(2, 4, 3)
```

٦ – ٣ إفرض N ومجموعة متراصة خطية A بها N عنصراً مخزنة فى الذاكرة . أكتب جزء برنامج فورتران يقوم بالتالى :

(أ) يحسب مجموع المربعات :

$$SUM = A_1^2 + A_2^2 + \cdots + A_n^2$$

(ب) يحسب حاصل الضرب

$$PROD = (1 - A_1)(1 - A_2) \ldots (1 - A_N)$$

(أ) أولا أجعل SUM = 0.0 قبل إجراء التجميع :

```
    SUM = 0.0
    DO 100 K = 1, N
        SUM = SUM + A(K)**2
100 CONTINUE
```

(ب) أولا إجعل PROD = 1.0 قبل إجراء الضرب . (هذا يشبه جعل SUM = 0.0 فى (أ)) .

```
    PROD = 1.0
    DO 100 K = 1, N
        PROD = PROD*(1.0 - A(K))
100 CONTINUE
```

٦ - ٤ أوجد الخرج فى البرنامج التالى :

```
      INTEGER A(10)
      DO 100 I = 1, 10, 2
            A(I) = 2*I - 3
            A(I + 1) = I**2 - 5
100   CONTINUE
      DO 200 I = 2, 10, 3
            A(I) = A(I + 2) - A(I)
200   CONTINUE
      WRITE(6, 10) A
10    FORMAT(1X, 5I8)
      STOP
      END
```

تعلن الجملة الأولى أن A مجموعة متراصة خطية بها 10 عناصر أول حلقة DO تنفذ ما يلى :

(١) أولا عندما I = 1 وهذا يعطى A(1) = −1 ، A(2) = −4

(٢) ثم عندما I = 3 . وهذا يعطى A(3) = 3 ، A(4) = 4

(٣) ثم عندما I = 5 وهذا يعطى A(5) = 7 ، A(6) = 20

(٤) ثم عندما I = 7 وهذا يعطى A(7) = 11 ، A(8) = 44

(٥) أخيراً عندما I = 9 وهذا يعطى A(9) = 15 ، A(10) = 76

وتنفذ حلقة DO الثانية ما يلى :

(١) أولا I = 2 تتغير قيمة A(2) كما يلى :

$$A(2) \leftarrow A(4) - A(2) = 4 - (-4) = 8$$

(٢) ثم I = 5 تتغير قيمة A(5) كما يلى :

$$A(5) \leftarrow A(7) - A(5) = 11 - 7 = 4$$

(٣) أخيراً I = 8 تتغير قيمة A(8) كما يلى :

$$A(8) \leftarrow A(10) - A(8) = 76 - 44 = 32$$

يطبع زوج الجمل WRITE-FORMAT قيم A كل خمسة منها على السطر فى حقول ذات عرض 8 حروف . من ثم سيظهر الخرج كما يلى :

```
   -1    8    3    4    4
   20   11   32   15   76
```

٦ - ٥ إفرض N ومجموعة متراصة خطية A بها عدد N عنصر مخزنة فى الذاكرة . أكتب جزءاً من برنامج فورتران يقوم بالآتى :

(أ) يبدل قيم A_1 و A_2 داخلياً يبدل قيم A_3 و A_4 داخلياً وهكذا ، (نفرض هنا أن N زوجية) .

(ب) يبدل دائرياً قيم A بحيث A_1 تحتوى القيمة الأصلية A_2 وتحتوى A_2 قيمة A_3 الأصلية ، وهكذا إلى أن تحتوى A_n قيمة A_1 الأصلية .

(أ) تذكر أولا أننا فبدل قيم X و Y باستخدام مخزن مؤقت T كما يلي :

```
T = X
X = Y
Y = T
```

وبذلك يكون لدينا جزء البرنامج :

```
      DO 100 K = 1, N, 2
         T = A(K)
         A(K) = A(K + 1)
         A(K + 1) = T
100   CONTINUE
```

(ب) أولا ، تخزن A_1 فى T مؤقتاً ، ثم ، بجعل A(1) = A(2) و A(2) = A(3)... A(N − 1) = A(N)
ثم أخيراً ، نجعل A(N) = T وبذلك يكون لدينا :

```
      T = A(1)
      NN = N − 1
      DO 100 K = 1, NN
         A(K) = A(K + 1)
100   CONTINUE
      A(N) = T
```

لاحظ مرة أخرى أننا نستخدم الجملة NN= N − 1 حيث أننا لانستطيع استخدام N − 1 كعامل .

برامج

٦ — ٦ إدرس مجموعة الأرقام x_1 ، x_2 ، ... x_n . يعرف كل من المتوسط الحسابى $\bar{x}$ والتباين ، والانحراف المعيارى σ بالآتى :

$$\bar{x} = (x_1 + x_2 + \cdots + x_n)/n = \left(\sum_{k=1}^{n} x_k\right)\Big/ n$$

$$\text{variance} = \frac{\sum_{k=1}^{n} (x_k - \bar{x})^2}{n}$$

$$\sigma = \sqrt{\text{variance}}$$

أفرض عدد N من الأرقام مخزنة فى مجموعة متراصة خطية A . أكتب جزء برنامج فورتران يخزن المتوسط ، التباين والانحراف المعيارى للأرقام فى AVE و VAR و SD على الترتيب (قارن مع مسألة ٥ — ٩ حيث استخدمنا المعادلة الثانية التباين) .

فيما يلى جزء البرنامج :

```
      SUM1 = 0.0
      DO 100 K = 1, N
         SUM1 = SUM1 + A(K)
100   CONTINUE
      AVE = SUM1/FLOAT(N)
      SUM2 = 0.0
      DO 200 K = 1, N
         SUM2 = SUM2 + (A(K) − AVE)**2
200   CONTINUE
      VAR = SUM2/FLOAT(N)
      SD = SQRT(VAR)
```

٦ ـ ٧ إدارة سلسلة محلات لها 6 فروع ، ولكل فرع نفس الإدارات العشر . تخزن المبيعات الأسبوعية للسلسلة فى مجموعة متراصة، (6 × 10) باسم SALES . تشير SALES (I, J) هنا إلى المبيعات الأسبوعية فى المحل رقم I والإدارة رقم J. أكتب جزء برنامج فورتران يقوم بالآتى :

(أ) يطبع مجموع المبيعات الأسبوعية لكل فرع .

(ب) يطبع مجموع المبيعات الأسبوعية لكل إدارة .

(ـحـ) يطبع مجموع المبيعات الأسبوعية للسلسلة بأكلها .

افترضنا هنا أن STORE و DEPT تم تعريفها بواسطة مجموعات متراصة خطية باستخدام الجملة :

```
DIMENSION STORE(6), DEPT(10)
```

(أ) نجمع مبيعات فرع معين I ، بعد أن نجعل STORE (I) = 0 فى البداية .

```
       WRITE(6, 10)
10     FORMAT('1', 5X, 'STORE', 5X, 'SALES'//)
       DO 99 I = 1, 6
              STORE(I) = 0.0
              DO 88 J = 1, 10
                     STORE(I) = STORE(I) + SALES(I, J)
88            CONTINUE
              WRITE(6, 20) I, STORE(I)
20            FORMAT(6X, I3, 5X, F9.2)
99     CONTINUE
```

(ب) نجمع مبيعات إدارة معينة I بعد أن نجعل DEPT(J) = 0 فى البداية .

```
       WRITE(6, 30)
30     FORMAT('0', 2X, 'DEPARTMENT', 3X, 'SALES'//)
       DO 77 J = 1, 10
              DEPT(J) = 0.0
              DO 66 I = 1, 6
                     DEPT(J) = DEPT(J) + SALES(I, J)
66            CONTINUE
              WRITE(6, 40) J, DEPT(J)
40            FORMAT(3X, I5, 5X, F9.2)
77     CONTINUE
```

(ـحـ) نستطيع أن نجمع القيم المخزنة فى المجموعة المتراصة SALES أو فى STORE أو فى DEPT (أو فى كل من الثلاثة للتأكد من النتائج) . نفعل ذلك أيضاً للمجموعة المتراصة STORE .

```
       CHAIN = 0.0
       DO 55 K = 1, 6
              CHAIN = CHAIN + STORE(K)
55     CONTINUE
       WRITE(6, 50) CHAIN
50     FORMAT('0', 'TOTAL WEEKLY SALES'/6X, F9.2)
```

٦ — ٨ إطبع أول 50 حد من أرقام فيبوناس بحيث يكون هناك خمسة أرقام فى السطر .

تذكر أن أرقام فيبوناس هى :

1, 1, 2, 3, 5, 8, 13, 21, . . .

أى أن كلا من الحدين الأولين يساوى 1 ونحصل على كل حد بعد ذلك بجمع الحدين السابقين .

2 = 1 + 1, 3 = 1 + 2, 5 = 2 + 3, 8 = 3 + 5, . . .

وبذلك ، نجعل :

J(1) = 1, J(2) = 1, and J(I) = J(I − 2) + J(I − 1)

عندما I > 2 . و

```
      DIMENSION J(50)
      J(1) = 1
      J(2) = 1
      DO 100 I = 3, 50
             J(I) = J(I - 2) + J(I - 1)
100   CONTINUE
      WRITE(6, 10)
 10   FORMAT('1', 15X, 'FIBONACCI SEQUENCE'//)
      WRITE(6, 20) J
 20   FORMAT(1X, 5(3X, I6))
      STOP
      END
```

لاحظ أن هناك خمسة مواصفات رقية فقط فى جملة FORMAT التى تحمل رقم السطر 20 وذلك بسبب طباعة كل خمسة أرقام على سطر .

٦ — ٩ (أ) إفرض أن مجموعة متراصة خطية B بها M عنصر . أكتب جزء البرنامج لإيجاد المكان J بحيث تحتوى B(J) على أكبر قيمة مطلقة مخزنة فى B (أنظر مثال ٦ — ١) .

(ب) إفرض أن A مصفوفة (M × N) مخزنة فى الذاكرة فعلا . اكتب جزء برنامج فورتران يقوم بالتالى :

(١) يجد المكان J بحيث تحتوى J على أكبر قيمة مطلقة فى العمود الأول .

(٢) يبدل الصف الأول مع الصف رقم J .

(من ثم ، ستحتوى A(1, 1) على القيمة الكبرى المطلقة فى العمود الأول) .

(أ) نجعل J = 1 و BIG = ABS(B(1)) ثم نقارن BIG مع بقية عناصر B إذا كانت BIG أقل من ABS(B(K)) ، فنجعل BIG = ABS(B(K) ونغير J إلى K . وفيما يلى البرنامج :

```
      J = 1
      BIG = ABS(B(1))
      DO 100 K = 1, M
          IF(BIG.GE.ABS(B(K))) go to 100
          BIG = ABS(B(K))
          J = K
100   CONTINUE
```

يحمل الدليل K فى البداية مساوياً 1 وتأخذ فى الاعتبار أن M يمكن أن تساوى 1 .

(ب) جزء البرنامج به إجراءان (١) إيجاد قيمة J (٢) إذا كانت J ≠ 1 تبديل الصف الأول بالصف رقم J .

```
C
C     FIND THE POSITION J WITH LARGEST ABSOLUTE VALUE
C
      J = 1
      BIG = ABS(A(1,1))
      DO 100 K = 1, M
          IF(BIG.GE.ABS(A(K,1)))GO TO 100
          BIG = ABS(A(K,1))
          J = K
  100 CONTINUE
      IF(J.EQ.1) GO TO 500
C
C     INTERCHANGE THE FIRST AND JTH ROWS
C
      DO 200 L = 1, N
          TEMP = A(1, L)
          A(1, L) = A(J, L)
          A(J, L) = TEMP
  200 CONTINUE
  500 ***********************************************
```

لاحظ التشابه بين الجزء الأول من البرنامج والجزء (أ) .

٦ — ١٠ اكتب جزء البرنامج الذى يضيف عنصر D فى المكان K من المجموعة المتراصة A(1) و A(2) و .... A(N) .

قبل أن تخصص D إلى A(K) يجب أن تحرك جزء المجموعة المتراصة A(K) و A(K + 1) و ..... A(N) إلى أسفل مكان واحد . يتم هذا بالبدء من نهاية المجموعة أى ، بأن تخصص A(N) إلى A(N + 1) ثم A(N — 1) إلى A(N) وهكذا حتى A(K) إلى A(K + 1) (أنظر قسم ٦ — ٨) لاحظ أولا أننا يجب أن تحرك N — K + 1 عنصراً ، وليس N — K حيث أننا نحرك أيضا A(K) . لنستخدم حلقة DO بدليل J = 1, 2,.......N — K + 1 . فيما يلى العلاقة بين الدليل J والأجراء الذى يتخذ :

قيمة J	الأجراء
1	A(N + 1) ← A(N)
2	A(N) ← A(N − 1)
3	A(N − 1) ← A(N − 2)
...	....
N − K + 1	A(K + 1) ← A(K)

فيما يلي جزء البرنامج :

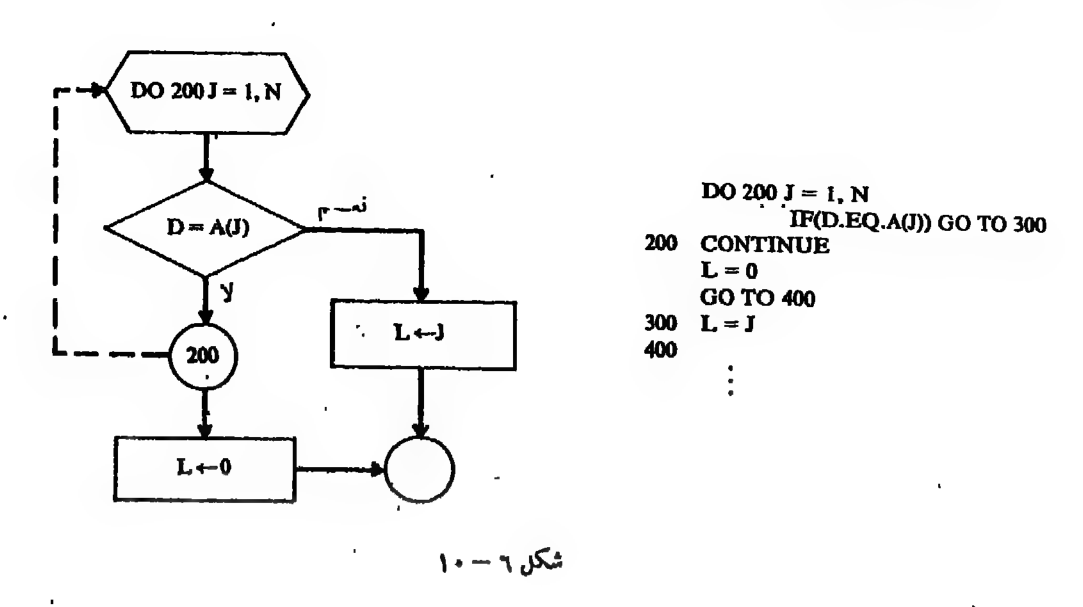
```
      NN = N - K + 1
      DO 100 J = 1, NN
         L = N - J + 1
         A(L + 1) = A(L)
100   CONTINUE
      A(K) = D
```

أو ببساطة

```
      NN = N - K + 1
      DO 100 J = 1, NN
         A(N - J + 2) = A(N - J + 1)
100   CONTINUE
      A(K) = D
```

٦ — ١١ (البحث الخطى) إذا أعطيت مجموعة متراصة A(1) و A(2) و .... A(N) وعنصر D ، أكتب جزء البرنامج الذى يجد المكان L للعنصر D عندما تقع قيمة D فى المجموعة المتراصة ، أو يجعل L = 0 عندما لا تقع قيمة D فى المجموعة المتراصة .

لاتوجد أى معلومات أخرى معطاة عن المجموعة المتراصة ، والطريقة الأكثر تلقائية لحل المسألة هى أن نقارن D مع كل عنصر من A أى ، نختبر إذا كانت D = A(1) ثم نختبر إذا كانت D = A(2) وهكذا . وتسمى هذه الطريقة البحث الخطى أو البحث المتعاقب (المتسلسل حيث أنها تبحث عن D باختبارها مع قيم عناصر A واحدة تلو الأخرى) (تناقش طرق أخرى البحث فى الفصل الثامن) . واضح أن هذه الطريقة تحتاج إلى عدد N من المقارنات لإيجاد D تحتاج عدد N من المقارنات عندما لا تكون D فى المجموعة المتراصة . يعرض الشكل ٦ — ١٠ خريطة سير العمليات النظام الحسابى (الخوارزم) وما يقابلها بالفورتران .




```
      DO 200 J = 1, N
         IF(D.EQ.A(J)) GO TO 300
200   CONTINUE
      L = 0
      GO TO 400
300   L = J
400
         :
```

شكل ٦ — ١٠

٦ — ١٢ (الفرز) . يقال أن المجموعة المتراصة مفروزة إذا كانت عناصرها مرتبة بأى نظام . أكتب جزء البرنامج الذى يفرز المجموعة المتراصة A(1) ، A(2) ، ..... A(N) بحيث تكون عناصرها مرتبة تصاعدياً .

تفرز A بالنظام الحسابى (الخوارزم) المعروف بـ « الفرز الفقاعى » وسوف (تناقش طرق أخرى للفرز فى الفصل الثامن) . نقارن أولا A(1) و A(2) ونرتبهما فى الترتيب المطلوب ، أى بحيث يكون A(1) < A(2) ثم نقارن A(2) و A(3) ونرتبهما بحيث يكون A(2) < A(3) ثم نقارن A(3) و A(4) ونرتبهما بحيث يكون A(3) < A(4) يكمل هذا إلى A(N—1) < A(N) — وبينما نلقى نظرة شاملة على كل العناصر « يطفو » العنصر الأكبر إلى المكان N .

تكرر العملية السابقة للعناصر A(1) ، A(2) ، .... A(N — 1) بعدما نلق نظرة شاملة على هذه العناصر ، يطفو العنصر الأكبر الثانى إلى الموقع N — 1 وهكذا بعد عدد N — 1 من المرات ستكون المجموعة المتراصة A مفروزة فى الترتيب التصاعدى .

ونذكر هنا شكل النظام الحسابى (الخوارزم) :

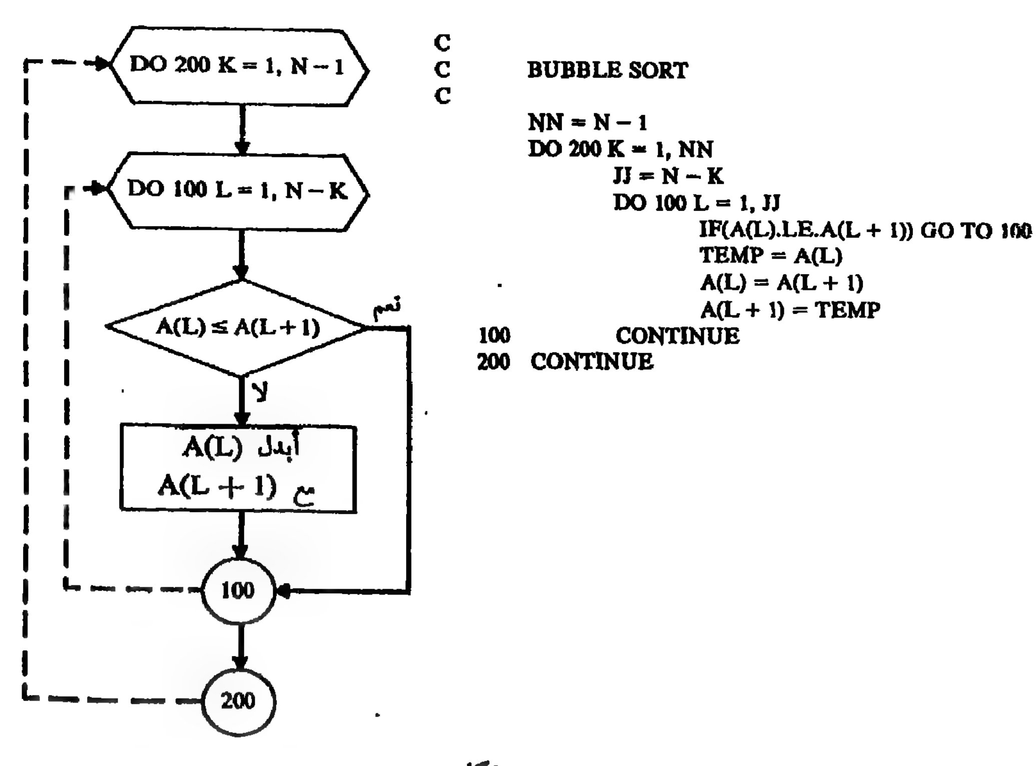
الخطوة ١ : لكل L = 1, 2, ...N — 1 قارن A(L) مع A(L+ 1) ورتبهما بحيث تكون A(L) < A(L + 1)

الخطوة ٢ : لكل L=1, 2, ... , N — 2 قارن A(L) مع A(L + 1) ورتبهما بحيث تكون A(L) < A (L +1)

....

الخطوة N — 1 : بالنسبة لـ L = 1 قارن A(L) مع A(L + 1) ورتبهما بحيث يكون A(L) < A(L + 1) .

لاحظ أن الإجراء الذى يتخذ متشابه فى كل خطوة . من ثم ، يمكن أن يكون لدينا حلقة DO محكومة بدليل K يأخذ القيم 1 ، 2 ، .... ، N—1 (تبعاً لعدد N—1 خطوة) ، وبداخل هذه الحلقة حلقة أخرى داخلية بدليل L يأخذ القيم 1 ، 2 ، .... ، N—K . يبين شكل ٦-١١ خريطة سير العمليات لهذا النظام الحسابى وترجمته للفورتران




```
C
C          BUBBLE SORT
C
           NN = N - 1
           DO 200 K = 1, NN
                 JJ = N - K
                 DO 100 L = 1, JJ
                       IF(A(L).LE.A(L + 1)) GO TO 100
                       TEMP = A(L)
                       A(L) = A(L + 1)
                       A(L + 1) = TEMP
   100           CONTINUE
   200     CONTINUE
```

شكل ٦ — ١١

٦ — ١٣ أكتب برنامجاً يقبل 25 رقماً صحيحاً موجباً مختلفاً N_1 ، N_2......N_{25} ويطبع ، على أسطر منفصلة ، كل زوج من الأرقام الصحيحة التى يكون مجموعها 75 . (ملحوظة : إذا كان 25 و 50 رقمين صحيحين من ثم فإننا نريد أما أن نطبع 25 و 50 أو نطبع 50 ، 25 ولكن ليس الإثنان معاً) .

$$N_1 + N_K = 75$$

أولا نقارن N_1 بكل من N_2 ، N_3....N_{25} لنرى إذا كانت

$$N_2 + N_K = 75$$

حيث $K > 1$ ثم نقارن N_2 بكل من N_3 ، N_4 ، ...N_{24} لنرى إذا كانت

$$N_J + N_K = 75$$

حيث $K > 2$ وهكذا . وبمعنى آخر لقيم J من 1 ، 2 ... 24 نتحقق لنرى إذا كانت

$$N_J + N_K = 75$$

لـ $K > J$. ويمكن عمل هذا بحلقات DO المتداخلة كما يلي :

```
      DIMENSION N(25)
      READ(5, 10) N
10    FORMAT(5I5)
      DO 100 J = 1, 24
           JJ = J + 1
           DO 200 K = JJ, 25
                IF(N(J) + N(K).NE.75) GO TO 200
                WRITE(6, 20) N(J), N(K)
20              FORMAT(1X, I5, 3X, I5)
200        CONTINUE
100   CONTINUE
      STOP
      END
```

٦ — ١٤ أدرس المعادلة متعددة الحدود :

$$y = 2x^4 - 5x^3 + 6x^2 - 8x + 9$$

(أ) اكتب جزء برنامج فورتران يخزن ويطبع قيمة y لقيم x حيث: $x = -10, -9, \ldots 10$ ، أي ، لقيم x الصحيحة من -10 إلى 10

(ب) اكتب جزء برنامج فورتران يخزن ويطبع قيمة y لقيم x حيث : $x = , -N, -N+1, \ldots, N$ حيث يقرأ N كبيان . (افرض $0 < N \leq 25$)

(أ) لاحظ أن هناك 21 قيمة لـ X من ثم نستخدم حلقة DO بدليل K: $K = 1, 2, \ldots 21$ وفيما يلي العلاقة بين X و K

قيمة X : -10 ، -9 ، -8 ، ..... 9 ، 10

قيمة K : 1 ، 2 ، 3 ، 20 ، 21

لاحظ أن : $X = -11 + K$. فيما يلي جزء البرنامج :

```
      DIMENSION Y(21)
      DO 100 K = 1, 21
           X = -11 + K
           Y(K) = (((2.*X - 5.)*X + 6.)*X - 8.)*X + 9.
           WRITE(6, 10) X, Y(K)
10         FORMAT(1X, F5.1, 2X, F8.1)
100   CONTINUE
```

(ب) لدينا الآن $2N + 1$ قيمة لـ X (لاحظ أن $2N + 1 \leq 51$ حيث $N \leq 25$) من ثم نستخدم حلقة DO بدليل K حيث : $K = 1, 2, \ldots 2N + 1$. هنا X و K . بينهما العلاقة $X = -N - 1 + K$.

```
      DIMENSION Y(51)
      READ(5, 20) N
20    FORMAT(I5)
      NN = 2*N + 1
      DO 88 K = 1, NN
            X = -N - 1 + K
            Y(K) = (((2.*X - 5.)*X + 6.)*X - 8.)*X + 9.
            WRITE(6, 30) X, Y(K)
30          FORMAT(1X, F5.1, 2X, F8.1)
88    CONTINUE
```

لاحظ أننا استخدمنا علاقة هورنر لحساب Y (أنظر قسم ٦ — ٨) .

٦ — ١٥ (التوزيع التكرارى) مجموعة بها 200 بطاقة تحتوى على رقم صحيح موجب K فى كل بطاقة ، حيث $K \leq 25$ اكتب البرنامج الذى يطبع الأرقام الصحيحة الموجبة من 1 إلى 25 وعدد المرات التى يظهر فيها الرقم الصحيح فى المجموعة .

نستخدم النظام الحسابى (الخوارزم) الآتى . تشير إلى أن N مجموعة متراصة خطية بها 25 عنصراً ، ونجعل N(K) تشير إلى عدد المرات التى ظهرت فيها قيمة K . بالتحديد ، نجعل فى البداية كل $N(K) = 0$ وكلما ظهرت قيمة K تزيد N(K) بواحد . على سبيل المثال . نجعل فى البداية $N(8) = 0$ فإذا ظهرت القيمة $K = 8$ وتكررت 14 مرة تصبح $N(8) = 14$ بعد قراءة كل البطاقات . وفيما يلى البرنامج :

```
      DIMENSION N(25)
C
C     SET EACH N(K) = 0
C
      DO 100 K = 1, 25
            N(K) = 0
100   CONTINUE
C
C     READ AND COUNT THE NUMBERS
C
      DO 200 J = 1, 200
            READ(5, 10) K
10          FORMAT(I5)
            N(K) = N(K) + 1
200   CONTINUE
C
C     PRINT THE DISTRIBUTION
C
      DO 300 K = 1, 25
            WRITE(6, 20) K, N(K)
20          FORMAT(6X, 2(I4, 5X))
300   CONTINUE
      STOP
      END
```

٦ — ١٦ اعتبر A مجموعة متراصة خطية بها N عنصر . نحصل على أول فرق DA للمجموعة المتراصة A بطرح كل عنصر ، من العنصر التالى له فى المجموعة المتراصة ماعداً العنصر الأخير

$$DA(K) = A(K + 1) - A(K)$$

حيث $K \le N-1$ لاحظ أن DA به $N-1$ عنصر . الفرق الثانى للمجموعة المتراصة D2A يعرف كالفرق الأول لـ DA وهكذا .

إفرض أن A مخزنة فى الذاكرة . أكتب جزء البرنامج الذى يجد أول فرق DA ، ثانى فرق D2A وثالث فرق D3A للمتتالية A :

2, 8, −3, 5, 9, −4, 8, 0 −8, 16

ثم يطبع A ، DA ، D2A ، D3A .

يمكن أن تبين الفروض كما يلى :

2	8	−3	5	9	−4	8	0	−8	16
6	−11	8	4	−13	12	−8	−8	24	
−17	19	−4	−17	25	−20	0	32		
36	−23	−13	42	−45	20	32			

أى ، الصف الثانى من الأرقام هو الفرق الأول ، والصف الثالث هو الفرق الثانى والصف الأخير هو الفرق الثالث .

وفيما يلى البرنامج .

```
C
C       FIND THE FIRST DIFFERENCE
C
        N = N - 1
        DO 100 K = 1, N
              DA(K) = A(K + 1) - A(K)
    100 CONTINUE
C
C       FIND THE SECOND DIFFERENCE
C
        N = N - 1
        DO 200 K = 1, N
              D2A(K) = DA(K + 1) - DA(K)
    200 CONTINUE
C
C       FIND THE THIRD DIFFERENCE
C
        N = N - 1
        DO 300 K = 1, N
              D3A(K) = D2A(K + 1) - D2A(K)
    300 CONTINUE
C
C       PRINT OUT THE ARRAYS
C
        WRITE(6, 10) A
        WRITE(6, 10) DA
        WRITE(6, 10) D2A
        WRITE(6, 10) D3A
     10 FORMAT(1X, 10(2X, F8.2))
```

إن لم تكن المجموعات المتراصة DA و D2A و D3A مطلوبة فيما بعد فى البرنامج ، يمكن أن نستخدم حلقات DO التكرارية المتداخلة .

حلقات DO التكرارية الضمنية

٦ – ١٧ أعد كتابة كل جملة فى حلقة DO الضمنية بحيث لايكون هناك أخطاء

```
READ(5, 10), (A(K) K = 1, 20)                          ( أ )
WRITE(6, 20), A(I, I), S, B(I), I = 1, N, 3            ( ب )
READ(5, 30) (A(J), J = 1, 7), B, (C(K), K = 1, M, L)   ( جـ )
WRITE(6, 40) (A(L, K + 3), K = 1, 7), L = 1, 5)        ( د )
```

```
READ(5, 10) (A(K), K = 1, 20)                          ( أ )
WRITE(6, 20) ((A(I, I), S. B(I)), I = 1, N, 3)         ( ب )
```

(جـ) لايوجد أخطاء .

(د) لانستطيع أن نستخدم K + 3 كدليل ، ولكن نستطيع أن نكتب

```
KK = K + 3
WRITE(6, 40) ((A(L, KK), KK = 4, 10), L = 1, 5)
```

٦ – ١٨ صف الخرج إذا نفذنا كل ثنائى من الجمل WRITE-FORMAT التالية :

```
   WRITE(6, 10) (A(K), K = 2, 25, 3)                   (أ)
10 FORMAT(1X, 5(3X, F10.2))
   WRITE(6, 10) (A(L), B, C(3, L), L = 1, 8, 2)        (ب)
10 FORMAT(1X, 5(3X, F10.2))
   WRITE(6, 10) (B(J), J = 1, 4), (B(K), K = 3, 10, 4) (جـ)
10 FORMAT(1X, 5(3X, F10.2))
```

(أ) لاحظ أن 2 هى القيمة الابتدائية لـ K وأن 3 هى معامل الزيادة و 25 قيمة النهاية. من ثم تطبع كل خمسة أرقام من A(2) و A(5) و A(8) و A(11) و A(14) A(17) و A(20) و A(23) على سطر منفصل وبذلك يظهر الخرج كما يلى :

```
A(2)    A(5)    A(8)    A(11)   A(14)
A(17)   A(20)   A(23)
```

(ب) لاحظ أن الحاسب ينفذ حلقة DO الضمنية أولا عندما L=1 ثم عندما L=3 ثم عندما L=5 وأخيراً عندما L=7. ومن ثم A(1) و B و C(3,1) و A(3) وB و C(3,3) و A(5) و B و C (3,5) و A(7) ، B و C(3,7) تطبع كل خمسة أرقام منها على سطر منفصل :

```
A(1)     B       C(3, 1)   A(3)      B
C(3, 3)  A(5)    B         C(3, 5)   A(7)
B        C(3, 7)
```

(جـ)

```
B(1)   B(2)   B(3)   B(4)   B(3)
B(7)
```

٦ - ١٩ صف الخرج عند تنفيذ ما يلى :

```
    WRITE(6, 20) ((A(J, K), J = 2, 8, 3), K = 1, 4)
20  FORMAT(IX, 5(3X, F10.2)
```

تعطى حلقة DO الضمنية الداخلية : A(2, K) و A(5, K) و A(8, K) ولما كانت K = 1, 2, 3, 4 فإننا نحصل على 12 قيمة تطبع كل خمسة منها على سطر . ويظهر الخرج بالصورة التالية :

```
A(2, 1)   A(5, 1)   A(8, 1)   A(2, 2)   A(5, 2)
A(8, 2)   A(2, 3)   A(5, 3)   A(8, 3)   A(2, 4)
A(5, 4)   A(8, 4)
```

٦ - ٢٠ مصفوفة A (N×N) مخزنة فى الذاكرة . أكتب جزء برنامج فورتران يطبع عناصر A الموجودة فوق القطر ، أى العناصر A(I, J) لكل قيم I < J.

نريد أن نطبع العناصر :

$$\begin{array}{llll} A_{12} & A_{13} & A_{14} \;\dots & A_{1N} \\ & A_{23} & A_{24} \;\dots & A_{2N} \\ & & \dots\dots\dots\dots & \\ & & & A_{N-1,N} \end{array}$$

يشير الدليل الأول I إلى صف العنصر ، ومن ثم يتغير من 1 إلى N — 1 . ويشير الدليل الثانى إلى العمود ، ومن ثم يتغير من I + 1 إلى N . لانستطيع أن نستخدم حلقة DO الضمنية :

```
WRITE(6, 10) ((A(I, J), J = I + 1, N), I = 1, N - 1)
```

حيث لايمكن أن تكون I + 1 ، N — 1 أدلة . ومع ذلك نستطيع أن نستخدم حلقة DO التكرارية مع حلقة DO التكرارية الضمنية :

```
      NN = N - 1
      DO 100 I = 1, NN
          K = I + 1
          WRITE(6, 10) (A(I, J), J = K, N)
10        FORMAT(1X, 8(3X, F10.2))
100   CONTINUE
```

مســـائل تكميلية

المجموعات المتراصة

٦ - ٢١ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل DIMENSION التالية :

(أ) `DIMENSION, J(8), INTEREST(5, 10),`

(ب) `DIMENSION A(4, 0, 3), B(6) C(9, 4)`

(جـ) `DIMENSION, A(M, N) XYZ(4, -8, 7)`

٦ – ٢٢ حدد عدد العناصر فى المجموعات المتراصة NAME و ID و X و Y تبعاً للجملة التالية

```
DIMENSION NAME(5, 300), ID(3, 5, 2), X(50), Y(4, 8)
```

٦ – ٢٣ باستخدام المجموعة المتراصة ID فى المسألة السابقة ٦ – ٢٢ ، صف الخرج إذا نفذنا الزوج التالى من الجمل :

```
   WRITE(6, 10) ID
10 FORMAT(1X, 8I8)
```

٦ – ٢٤ باستخدام المجموعة المتراصة Y فى المسألة ٦ – ٢٢ ، أكتب زوج الجمل WRITE-FORMAT الذى يطبع الصفوف الأربعة من Y على السطور الأربعة الأولى من صفحة الطباعة على الترتيب .

٦ – ٢٥ أوجد الخرج الخاص بكل جزء برنامج مما يلى :

(أ)

```
    DIMENSION ID(5)
    DO 100 K = 1, 5
        ID(K) = 2*K
100 CONTINUE
    ID(2) = ID(2) + ID(3)
    ID(5) = ID(5) - ID(4)
    WRITE(6, 10) ID
 10 FORMAT(1X, 5I8)
```

(ب)

```
    INTEGER X(6)
    DO 100 K = 1, 6, 2
        X(K) = 3*K
        X(K + 1) = K + 2
100 CONTINUE
    DO 200 J = 1, 6, 3
        X(J) = X(J) + X(J + 1)
200 CONTINUE
    IF(X(2).LT.7) X(2) = X(3)
    WRITE(6, 20) X
 20 FORMAT(1X, 6I8).
```

٦ – ٢٦ إفرض مجموعة متراصة خطية A بها N عنصر مخزنة فى الذاكرة . أكتب جزء البرنامج الذى يقوم بالتالى :

(أ) يبدل A_1 و A_2 ما إذا كانت A_1 أكبر من A_2 فقط ، يبدل A_3 وA_4 ما إذا كانت A_3 أكبر من A_4 فقط ، وهكذا (نفترض هنا N زوجية)

(ب) يبدل دائرياً قيم A بحيث A_2 تحتوى القيمة الأصلية A_1 و A_3 تحتوى القيمة الأصلية لـ A_2 وهكذا ، وتحتوى A_1 القيمة الأصلية لـ A_N.

حلقات DO التكرارية الضمنية

٦ – ٢٧ صف الخرج إذا نفذنا زوج الجمل WRITE-FORMAT التالى :

(أ)

```
   WRITE(6, 10) (A(N), N = 4, 28, 5)
10 FORMAT(1X, 3F15.2)
```

(ب)

```
   WRITE(6, 20) (J(L), K, N(3, L), L = 2, 13, 3)
20 FORMAT(1X, 4I10)
```

٦ — ٢٨ صف الخرج إذا نفذنا زوج الجمل WRITE-FORMAT التالى :

(أ)
```
   WRITE(6, 30) (A(M), M = 1, 5), X, (B(N), N = 1, 7, 3)
30 FORMAT(1X, 4F15.2)
```
(ب)
```
   WRITE(6, 40) ((A(M, N), M = 1, 5), N = 1, 7, 3)
40 FORMAT(1X, 4F15.2)
```
(جـ)
```
   WRITE(6, 50) ((A(M, N), N = 1, 7, 3), M = 1, 5)
50 FORMAT(1X, 4F15.2)
```

بـرامج

٦ — ٢٩ أدى فصل به 35 طالباً إمتحان تتراوح درجاته ما بين 0 و 100 أكتب برنامج فورتران يجد : (أ) متوسط الدرجات ، (ب) عدد الطلبة الراسبين أى الحاصلين على درجات أقل من 60 درجة و (جـ) عدد الطلبة الممتازين أى ، الحاصلين على 100درجة .

٦ — ٣٠ يؤدى الطالب سبعة اختبارات (تتراوح درجاتها ما بين 0 و 100) ، ودرجته النهائية هى متوسط أعلى ست درجات فى الاختبارات . باعتبار أن الدرجات السبع للاختبار قد تم تثقيبها على بطاقة بيانات . اكتب جزء برنامج الفورتران الذى يجد درجة الطالب النهائية . (تلميح . اطرح أصغر درجة من مجموع الدرجات السبع SUM) .

٦ — ٣١ يستثمر رجل رأس ماله فى حساب توفير يربح 7 فى المائة كل سنة ولمدة 10 سنوات بفائدة مركبة . اكتب البرنامج الذى يطبع ما وصل إليه حسابه كل سنة لمدة 20 سنة (لاحظ أنه ليست هناك استثمارات إضافية فى آخر 10 سنوات)

٦ — ٣٢ سلسلة إدارة فروع بها 6 محلات ، كل فرع به نفس عدد الإدارات وهى 12 . المبيعات الأسبوعية للسلسلة تخزن فى مجموعة متراصة $(6 \times 12 \times 7)$ باسم SALES بحيث تشير SALES(I, J, K) إلى المبيعات فى الفرع I الإدارة J فى اليوم K . كتب جزء برنامج فورتران يقوم بالتالى :

(أ) يطبع مجموع المبيعات الأسبوعية لكل فرع .
(ب) يطبع مجموع المبيعات الأسبوعية لكل إدارة .
(جـ) يطبع مجموع المبيعات الأسبوعية للسلسلة بأكملها .
(قارن مع مسألة ٦ — ٧) .

٦ — ٣٣ سلسلة إدارة محلات بها 14 فرع (مرقة من 1 إلى 14) وكل فرع به نفس عدد الإدارات وهى 8 إدارات . يقدم كل فرع بطاقة بيانات كل أسبوع إلى المكتب الرئيسى مثقب عليها رقم الفرع فى الأعمدة 1 إلى 3 ثم تثقب 8 أرقام تمثل المبيعات الأسبوعية فى كل إدارة . أكتب جزء برنامج فورتران يقوم بالتالى :

(أ) يقرأ البيانات فى المجموعة المتراصة SALES بحيث تحتوى SALES(I, J) على المبيعات الأسبوعية فى المحل I و الإدارة J . (إفرض أن بطاقات البيانات ليست مرتبة بأى ترتيب معين) .
(ب) يطبع مجموع المبيعات لكل فرع .
(جـ) يطبع رقم الفرع صاحب أكبر مبيعات .

٦ — ٣٤ إذا أعطيت أرقاماً صحيحة موجبة I و J وكانت أرقام فيبوناسى المعممة لـ $K > 2$ تعرف بما يلى :

$$JFIB(1) = I, \quad JFIB(2) = J, \quad \text{and} \quad JFIB(K) = JFIB(K-2) + JFIB(K-1)$$

(أنظر مسالة ٦ — ٨) . أكتب البرنامج الذى يقرأ I و J ويطبع أول 30 حد من السلسلة بحيث يطبع ثلاثة أرقام فى السطر .

٦ — ٣٥ اكتب برنامج الفورتران الذى :

(أ) يقبل 25 رقماً صحيحاً موجباً ويقرر ما إذا كانت أى رقمين من الأرقام الصحيحة مجموعها 15 .

(ب) يقبل 15 رقماً صحيحاً موجباً ويجد الرقم التالى لأصغر رقم والرقم التالى لأكبر رقم .

(جـ) يقبل 25 رقماً صحيحاً موجباً ويجد أكبر رقم زوجى ، أن لم يجده يطبع THERE ARE NO EVEN INTEGERS

(د) يقبل 25 رقماً صحيحاً موجباً ويحسب عدد الأرقام الزوجية منها ، وكذا عدد الأرقام الفردية .

٦ — ٣٦ ثقبت درجات لجنة اختبار دخول كلية لعدد 450 طالباً على بطاقات ، درجة واحدة فى كل بطاقة . (تتراوح تلك الدرجات ما بين 200 و 800) . اكتب البرنامج الذى يجد عدد الدرجات بين 200 و 300 وبين 300 و 400 وهكذا .

٦ — ٣٧ اكتب البرنامج الذى يقرأ المعاملات a_1 ، a_2 ، ...., a_{n-1} لكثيرة الحدود .

$$p(x) = a_1x^n + a_2x^{n+1} + \cdots + a_nx + a_{n+1}$$

ويخزنها فى المصفوفة الخطية COEF ويطبع قيم $p(x)$ لقيم $x = -5$ و -4.5 و -4 و ... و 4.5 و 5 أى لقيم x من -5 إلى 5 بخطوات مقدارها 0.5 . استخدم طريقة هورنر لإيجاد قيمة $p(x)$ أى :

$$p(x) = ((\ldots((a_1x + a_2)x + a_3)x + \cdots + a_{n-1})x + a_n)x + a_{n+1}$$

٦ — ٣٨ (نظرية المجموعات) إفرض JSET و KSET مجموعات متراصة خطية كل منها يحتوى على أرقام صحيحة موجبة مختلفة عن بعضها .

(أ) اكتب جزء البرنامج الذى يخزن فى INTER العناصر المشتركة بين JSET و KSET أى تلك العناصر الموجودة فى JSET و KSET أيضاً .

(ب) اكتب جزء البرنامج الذى يخزن فى JUNION العناصر كلها فى JSET و KSET أى تلك العناصر فى JSET أو KSET (أو كليهما) .

٦ — ٣٩ قائمة جرد البضائع فى مستودع لأجهزة TV مخزنة فى مجموعتين متراصتين صحيحتين ID و AMOUNT كما يلى . يعطى كل مدخل فى ID رقم طراز TV (رقم من ست خانات) ، ويعطى المدخل المقابل فى AMOUNT عدد التليفزيونات من هذا الطراز فى المستودع . يقدم كل أمر شحن الوارد أو الصادر مجموعة من بطاقات البيانات (لها بطاقة خلفية) ثقب عليها رقم الطراز فى الأعمدة من 1 إلى 8 وتثقبت كية الصنف المشحون فى الأعمدة من 11 إلى 20 . (حيث يكون العدد موجباً إذا كانت الشحنة واردة إلى المستودع ويكون سالباً إذا كانت مصدرة من المستودع) . أكتب البرنامج الذى يعدل قائمة الجرد بعد كل أمر شحن . يجب أن يطبع أيضاً :

INSUFFICIENT QUANTITY MODEL NUMBER XXXXXX

إذا كان الأمر الصادر من المستودع لصنف يتجاوز العدد فى الرصيد .

٦ — ٤٠ A مصفوفة (N × N) مخزنة فى الذاكرة .

(أ) اكتب البرنامج الذى يجمع العناصر فوق القطر الرئيسى ، أى العناصر A(I, J) بحيث I < J .

(ب) اكتب البرنامج الذى يجمع العناصر أسفل القطر الرئيسى ، أى ، العناصر A(I, J) بحيث I > J

٦ - ٤١ تحتوى (لعبة التصويب بالكرة) على مجموعة متراصة خطية A لعدد الأوتاد المضروبة فى كل ضربة . أكتب جز ' رنامج الذى يخزن النتيجة بعد كل دورة فى مجموعة متراصة B وعلى ذلك B(10) تعطى النتيجة النهائية . اختبر البرنامج بالبيانات التالية :

(أ) 4, 5, 10, 10, 6, 4, 7, 3, 8, 0, 7, 3, 10, 8, 1, 10, 7, 2

(ب) 6. 4. 7, 2, 10, 7, 3, 10, 5, 4, 3, 0, 10,.10, 8, 2, 7

٦ — ٤٢ (طريقة سيف) اكتب البرنامج الذى يطبع الأعداد الأولية التى أقل من 400 كما يلى :

(أ) إجعل A مجموعة متراصة خطية بها 400 عنصر .

(ب) خزن K فى A(K) أى ، إجعل A(1) = 1 و A(2) = 2 و ...400 = A(400) .

(ـ) اطبع A(2) ثم ضع كل مضاعفات 2 مساوية للصفر .

(د) اطبع العنصر التالى والذى لايساوى صفراً فى المجموعة المتراصة وهو A(3) ، ثم ضع كل مضاعفات 3 مساوية للصفر .

(هـ) كرر العمليات السابقة طالما أن $A(K) \leq 20 = \sqrt{400}$.

(و) عند : $A(K) > 20$ اطبع كل العناصر غير الصفرية فى A .

اجابات للمسائل التكميلية المختارة

٦ — ٢١ (أ) يجب ألا تكون هناك فصلة بعد DIMENSION ولا فى النهاية .

(ب) يجب أن تكون هناك فصلة بعد B(6) . أيضاً لا يمكن أن يكون الصفر دليلا .

(ـ) يجب ألا تكون هناك فصلة بعد DIMENSION ؛ يجب أن تكون هناك فصلة قبل XYZ ؛ ولا يمكن أن تكون M و N و 8 — أدلة .

٦ — ٢٢ 1500 و 30 و 50 و 32 .

٦ — ٢٣ حيث أن اسم المجموعة المتراصة هو الذى يظهر فقط ، ستطبع ID بترتيب التخزين أى ستطبع بطريقة الأعمدة من الصفحة الأولى ثم بطريقة الأعمده من الصفحة الثانية مع كل ثمانية عناصر على سطر .

٦ — ٢٤

```
   WRITE(6, 20) ((Y(I, J), J = 1, 8), I = 1, 4)
20 FORMAT('1', 8F15.2/1X, 8F15.2/1X, 8F15.2/1X, 8F15.2)
```

٦ — ٢٥ (أ) الأرقام الصحيحة 2 و10 و6 و 8 و2 ستطبع مضبطة من الطرف الأيمن فى حقول بعرض 8 خانات .

(ب) الأرقام الصحيحة 6 و 9 و9 و 20 و 15 و 7 ستطبع فى حقول بعرض 8 خانات .

٦ — ٢٦ (أ)

```
   DO 99 K = 1, N, 2
      IF(A(K).LE.A(K + 1)) GO TO 99
      T = A(K)
      A(K) = A(K + 1)
      A(K + 1) = T
99 CONTINUE
```

(ب)

```
   T = A(N)
   NN = N - 1
   DO 88 K = 1, NN
      A(N - K + 1) = A(N - K)
88 CONTINUE
   A(1) = T
```

٦ – ٢٧ (أ) A(4) و A(9) و A(14) على سطر واحد و A(19) و A(24) على السطر الثاني .
(ب) J(2) و K و N(3, 2) ، J(5) على سطر واحد ؛ K و N(3, 5) و J(8) و K على السطر التالي ، و N(3, 8)
J(11) و K و N(3, 11) على السطر الثالث .

٦ – ٢٨ (أ) A(1) و A(2) و A(3) و A(4) على سطر واحد ؛ A(5) و X و B(1) و B(4) على السطر التالي و B(7) على السطر الثالث .
(ب) A(1,1) و A(2,1) و A(3,1) و A(4,1) على سطر واحد A(5, 1) و A(1, 4) و A(2, 4) و A(3, 4) السطر التالي A(1, 7) و A(2, 7) A(4,4) و A(5,4) على السطر الثالث ؛ A(3, 7) و A(4, 7) و A(5, 7) على السطر الرابع .
(جـ) A(1, 1) و A(1, 4) و A(1, 7) و A(2, 1) على سطر واحد A(2, 4) و A(2, 7) و A(3, 1) و A(3, 3) على السطر التالي ؛ A(3, 7) و A(4, 1) و A(4, 4) و A(4, 7) وعلى السطر الثالث ؛ A(5, 1) و A(5, 4) و A(5, 7) على السطر الرابع .

الفصل السابع

الدوال والبرامج الصغيرة الفرعية

٧ — ١ مقدمـــــة

قد تظهر عدة مشاكل كلما ازدادت البرامج تعقيداً أو أصبحت أكثر شمولا فمثلا:

١ — يصعب تصميم النظام الحسابي (الخوارزم) لحل المسائل المعقدة .

٢ — حتى إذا كان النظام الحسابي (الخوارزم) معروفاً فإن تنفيذه فى الفورتران يصبح أكثر صعوبة وذلك لطول البرنامج .

٣ — كلما أصبحت البرامج أطول وأعقد ، يصبح اكتشاف الأخطاء أكثر صعوبة . وتكون البرامج الأطول أكثر تعرضاً لخطأ الذى يصعب تجنبه .

٤ — نحتاج إلى المزيد من توثيق البرامج حتى تصبح أسهل فهماً لأولئك الذين سيقرؤن ويستخدمون البرامج .

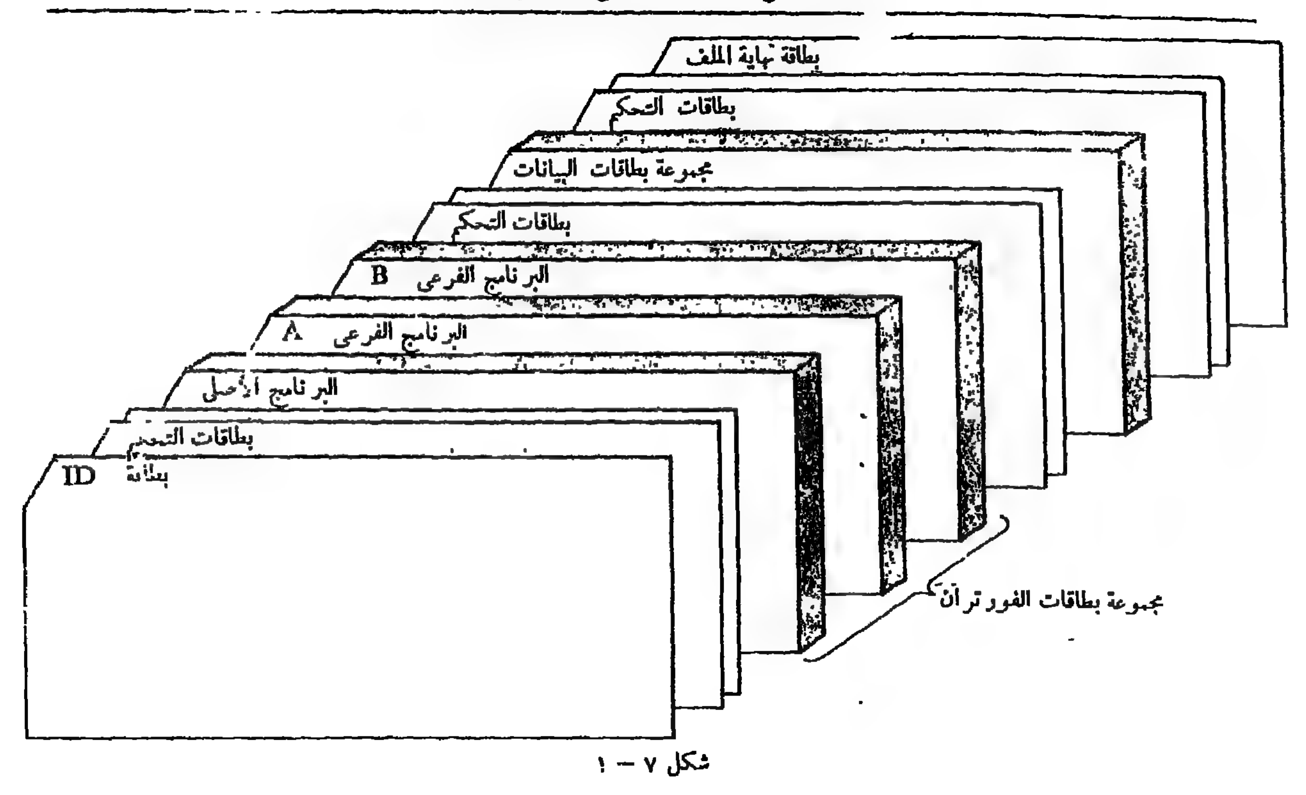
٥ — قد نحتاج إلى مهام مماثلة فى أجزاء متعددة من البرنامج . إذا نفذت مفردة أى واحدة تلو الأخرى فإنها تجعل المسألة المعقدة أكثر إرباكاً .

يمكن تخفيف هذه المشكلة باستخدام البرامج الفرعية . فالبرنامج الفرعى هو برنامج كامل ومستقل يمكن استخدامه (أو استدعاؤه) بواسطة البرنامج الأساسى أو بواسطة برامج فرعية أخرى . ويستقبل البرنامج الفرعى من البرنامج الأصلى (أو الداعى) قيم (تسمى خلاصات) وذلك لتنفيذ الحسابات ، وبعد ذلك تعيد (RETURNs) النتيجة أو النتائج إلى البرنامج الأصلى (أو الداعى) .

فى الفصل الثانى ، ناقشنا بعض الدوال المكتبية مثل ABS و SQRT و EXP و ALOG إلخ . إن هذه الدوال فى الحقيقة هى برامج فرعية كتبت خصيصاً لحل هذه البرامج وأدمجت مع بعضها فى نص الفورتران نظراً لكثرة استخدامها فى مواقف عديدة . وتسهيلا لمهمة كل مستخدم لهذه البرامج الفرعية وحتى لا يضطر لكتابتها فإن مترجمات الفورتران تمده بهذه الدوال المكتبية داخلياً . ومن الواضح ، أنه لا يمكن أن تحتوى مكتبة الفورتران على كل الدوال أو البرامج الفرعية المستخدمة فى كل مجالات التطبيق ، وعلى المستفيدين أن يكتبوا برامجهم الفرعية الخاصة كلما كانت هناك حاجة لذلك . ولقد خصص هذ االفصل لمناقشة إعداد واستخدام مثل هذه البرامج الفرعية .

نذكر ترتيب حزمة الفورتران الموضحة فى الشكل ١ — ٣ فى قسم ١ — ٣ . يجب أن تتكون مجموعة الفورتران من البرنامج الأساسى ، بصحبة كل البرامج الفرعية . وعادة نضع البرنامج الأساسى أولا بعد ذلك البرامج الفرعية . والشكل ٧ — ١ يوضح مجموعة الفورتران بها برنامجان فرعيان .

تقع البرامج الفرعية فى مرتبتين أساسيتين : البرامج الفرعية FUNCTION والبرامج الفرعية SUBROUTINE وسيوضح فى الأقسام القليلة التالية أوجه الشبه والاختلاف بين هذين النوعين من البرامج الفرعية . مع أن كلا النوعين من البرامج الفرعية يمكن ان يستقبل قيمة أو أكثر من البرنامج الأصلى (أو الداعى) فالبرنامج الفرعى FUNCTION يعود بنتيجة واحدة فقط ، فى حين أن SUBROUTINE عادة يعود بأكثر من نتيجة واحدة .



شكل ٧ — ١

٧ — ٢ الدوال كبرامج فرعية

نلاحظ أن كلا من الدوال المكتبية ABS و SQRT و SIN و ALOG ، إلخ تحسب قيمة كلما أعطيت الخلاصة المضبوطة . افرض ، على سبيل المثال ، إذا ظهرت

SQRT(A**2 + B**2)

فعندما ينفذ التعبير الحسابى ، فإن الحاسب يقوم بالتالى :

١ — ينادى على البرنامج الفرعى SQRT

٢ — يحسب القيمة الحقيقية $A^2 + B^2$ ويستخدم هذه القيمة كخلاصة فى البرنامج الفرعى .

٣ — يحسب البرنامج الفرعى الجذر التربيعى للخلاصة المعطاة وهى $(A^2 + B^2)$ ويخصص هذه القيمة الحقيقية المتغير SQRT فى البرنامج الفرعى

٤ — يعيد قيمة SQRT من البرنامج الفرعى ثانية إلى البرنامج الأصلى (أو الداعى) ، وتعوض هذه القيمة SQRT (A**2 + B**2)

لاحظ أن هذا البرنامج الفرعى له قيمة مرتبطة باسمه SQRT . البرامج الفرعية من هذا النوع تسمى الدوال كبرامج فرعية FUNCTION

نعطى الآن مثالين للبرامج الفرعية FUNCTION سنشير إليهما بعد ذلك عندما نذكر قواعد كتابة البرامج الفرعية .

البرنامج الفرعى الأول

ادرس مسألة إيجاد أكبر رقم من ثلاثة أرقام A و B و C . افرض أن A و B و C مخزنة فى الذاكرة و BIG متغير حقيقى . سوف يستخدم لتخزين القيمة الكبرى ، شكل ٧ — ٢ (أ) يبين جزء الفورتران الذى ينتهى بعد حساب BIG

```
BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
STOP
END
```

(أ)

```
FUNCTION BIG(A, B, C)
BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
RETURN
END
```

(ب)

شكل ٧ — ٢

افرض أننا نريد أن نكتب هذا الجزء من الفورتران كبرنامج فرعى (دالة) FUNCTION يجب أن تكون أول جملة هى جملة تعريف FUNCTION التى تخبر المترجم أن هذا هو برنامج فرعى FUNCTION . تعطى جملة تعريف FUCTION اسم البرنامج الفرعى وخلاصاته . وسنطلق على برنامجنا الفرعى BIG وله ثلاثة معاملات A و B و C ولذلك يكون أول جملة فى البرنامج الفرعى .

FUNCTION BIG(A, B, C)

نعتبر أن ... المعاملات فى البرنامج الفرعى قد تم تعريفها . وعلاوة على ذلك ، وحيث أن هذا البرنامج الفرعى FUNCTION سيستخدم بواسطة أى برنامج داعى ، فإن التنفيذ لا يتوقف بجملة STOP بعد الحسابات ، ولكن سيحتوى على جملة RETURN التى تنقل التحكم مرة ثانية إلى البرنامج الداعى . يعرض شكل ٧ — ٢ (ب) جزء البرنامج مكتوب كبرنامج فرعى FUNCTION كامل . لاحظ التشابه بين البرنامج الفرعى وجزء الفورتران شكل ٧ — ٢ (أ) .

البرنامج الفرعى الثانى

الدالة $k!$ (مضروب k) تعرف بالتالى :

$$k! = \begin{cases} 1\cdot 2\cdot 3\cdot \ldots \cdot (k-1)\cdot k, & \text{if } k \geq 1 \\ 1 & \text{if } k = 0 \end{cases}$$

على سبيل المثال $0! = 1$ ، $4! = 1 \times 2 \times 3 \times 4 = 24$ ، $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$

إدرس مسألة حساب $k!$ لأى رقم صحيح موجب k باعتبار k مخزنة فى الذاكرة و IFACT هى متغير صحيح يستخدم لتخزين $k!$. ويعطى شكل ٧ — ٣ (أ) جزء الفورتران الذى ينتهى بعد انتهاء حساب IFACT . من ناحية أخرى ، يعرض شكل ٧ — ٣ (ب) كيف نحصل على IFACT من برنامج فرعى FUNCTION لاحظ أن هناك جملتى RETURN تناظر جملتى STOP

```
      IFACT = 1
      IF(K.EQ.0) STOP
      DO 10 J = 1, K
         IFACT = IFACT*J
   10 CONTINUE
      STOP
      END
```

(أ)

```
      FUNCTION IFACT(K)
C
C     COMPUTES K FACTORIAL
C
      IFACT = 1
      IF(K.EQ.0) RETURN
      DO 10 J = 1, K
            IFACT = IFACT*J
   10 CONTINUE
      RETURN
      END
```

(ب)

شكل ٧ — ٣

نلخص الآن النقاط الرئيسية فى كتابة برامج دالة فرعية FUNCTION

١ – يجب أن تكون أول جملة من البرنامج الفرعى جملة تعريف FUNCTION ويكون لها الشكل التالى :

FUNCTION NAME(param1, param2, . . . , param*n*)

أى أن كلمة FUNCTION يتبعها اسم الدالة NAME ويتبع NAME المعاملات التى يفصلها عن بعضها بواسطة فصلات وتحاط بأقواس . ويجب أن تكون هذه المعاملات أسماء متغيرات بدون أدلة (أو أسماء مجموعات ترابية انظر قسم ٧ – ٦) ، مع فرض تعريفهم مسبقاً بالنسبة للبرنامج الفرعى .

٢ – تسمية البرنامج الفرعى FUNCTION تتبع نفس القواعد الخاصة بتسمية المتغيرات ؛ أى ، يجب أن تتكون من حرف واحد إلى ستة حروف أبجدية رقية مع مراعاة أن يكون الحرف الأول أبجدياً . يجب أن يظهر اسم البرنامج الفرعى FUNCTION NAME (بدون معاملاته) أيضاً بأى طريقة فى البرنامج الفرعى (أى يظهر إما على الجانب الأيسر من جملة تخصيص أو فى جملة إدخال) قبل تنفيذ أى جملة RETURN (وبذلك يجب أن يتبع البرنامج الفرعى FUNCTION مفهوم النوع لأسماء المتغيرات) .

٣ – البرنامج الفرعى هو برنامج كامل ومستقل :

(ا) كامل لأنه يحتوى على تعريفات النوع المناسبة (مثل INTEGER و REAL و DIMENSION إلخ ولأنه يمكن أن يستخدم الدوال المكتبية وبرامج فرعية أخرى وتحتوى على جملة END

(ب) مستقل لأن كل أسماء المتغيرات وكل أرقام الجمل المستخدمة داخليا بالنسبة للبرنامج الفرعى (ماعدا المعاملات) غير معروفة خارج نطاقه . وبذلك ، يمكن أن تستخدم نفس أسماء المتغيرات ونفس أرقام الجمل فى البرامج الفرعية المختلفة والبرامج التى تستدعيها .

٤ – وسيلة الاتصال الوحيدة بين البرنامج الداعى والبرنامج الفرعى هى من خلال المعاملات (انظر قسم ٧ – ٣) .

يجب أن يحتوى كل برنامج فرعى FUNCTION على معامل واحد على الأقل وجملة RETURN واحدة وهى تنقل التحكم ثانية إلى البرنامج الداعى .

٧ – ٣ استدعاء البرامج الفرعية FUNCTION

يستحضر (أو يستدعى) البرنامج الفرعى FUNCTION بنفس طريقة أى دالة مكتبية . بالتحديد فى تلك الأماكن التى يراد فيها قيمة الدالة نكتب :

NAME(arg1, arg2, . . . , arg*n*)

أى ، نكتب اسم الدالة NAME متبوعاً بتعبيرات مناسبة لخلاصاتها التى تفصل عن بعضها بواسطة فصلات وتحاط بأقواس .

ليس لزاماً أن تكون أسماء الخلاصات فى جملة الاستدعاء هى نفسها الموجودة فى المعاملات المناظرة فى جملة تعريف FUNCTION . فى الحقيقة ، يمكن أن تكون الخلاصات ثوابت أو متغيرات مفردة أو تعبيرات حسابية أو حتى أسماء دوال مكتبية أو برامج فرعية أخرى كما فى الدوال المكتبية . أهم شئ يجب أن تتذكره هو أن الخلاصات يجب أن تتفق بطريقة واحد مقابل واحد فى الترتيب ونوع البيانات للمعاملات المناظرة فى جملة تعريف FUNCTION على سبيل المثال ، ادرس البرنامج الفرعى FUNCTION المسمى BIG المذكور سابقاً وجملة تعريفه هى :

FUNCTION BIG(A, B, C)

لاحظ أن المطلوب ثلاث خلاصات حقيقية من أجل استدعاء BIG بحيث تكون الجمل التالية كلها صحيحة لغوياً .

```
GRADE = BIG(T1, T2, T3)
GRADE = FINAL/3.0 + (2.0/3.0)*BIG(T1, 60.0, SQRT(A))
GRADE = 0.5*(FINAL + BIG(S(I, 1), S(I, 2), Z(I))
```

وفيما يلى نوضح المناظرة واحد مقابل واحد بين خلاصات الجملة الثانية ومعاملات البرنامج الفرعى :

قائمة بالخلاصات	T1,	60.0,	SQRT(A)
	↓	↓	↓
قائمة بالمعاملات	A,	B,	C

هناك نقطة واحدة خفية يجب أن نفهمها عند استخدام البرامج الفرعية . وهى أن المعاملات المستخدمة فى جملة تعريف الدالة FUNCTION تسمى المتغيرات الزائفة وذلك لأنه فى الحقيقة لا يحدد لها أماكن تخزين . وبالتحديد ، فهى تستخدم عناوين الخلاصات المقابلة فى البرنامج الداعى عند تنفيذ البرنامج الفرعى . على سبيل المثال ، فى جملة الاستدعاء .

```
GRADE = BIG(T1 + T2, C, 60.0)
```

ستستخدم عناوين الخلاصات T1 + T2 و C و 60.0 بدلا من المعاملات A و B و C عند تنفيذ البرنامج الفرعى BIG .

ملاحظة : لاحظ أننا نستخدم المصطلح معاملات للمتغيرات فى البرنامج الفرعى التى تظهر فى جملة التعريف والمصطلح خلاصات القيم المناظرة لها فى البرنامج الداعى .

مثال ٧ – ١

(ا) يعرف المعامل ذو الحدين $\binom{n}{i}$ للأرقام الصحيحة الموجبة n و i (حيث $i \le n$) بما يلى :

$$\binom{n}{i} = \frac{n!}{(n-i)!i!};\ \text{for example,}\ \binom{8}{3} = \frac{8!}{5!3!} = \frac{1\cdot2\cdot3\cdot4\cdot5\cdot6\cdot7\cdot8}{1\cdot2\cdot3\cdot4\cdot5\cdot1\cdot2\cdot3} = \frac{6\cdot7\cdot8}{1\cdot2\cdot3} = 56$$

اكتب البرنامج الذى يقبل أرقاماً موجبة N و I (حيث I ≤ N) يطبع القيمة $\binom{N}{I}$

باستخدام البرنامج الفرعى IFACT فى القسم السابق فإن برنامج الفورتران التالى يعطينا المعامل ذا الحدين المطلوب

```
C     PROGRAM CALCULATING BINOMIAL COEFFICIENT
      INTEGER UP, DOWN
      READ(5, 10) N, I
   10 FORMAT(2I8)
      UP = IFACT(N)
      DOWN = IFACT(N - I)*IFACT(I)
      IBINQ = UP/DOWN
      WRITE(6, 20) N, I, IBINQ
   20 FORMAT(6X, 'N = ', I5, 5X, 'I = ', I5, 5X,
     1         'BINOMIAL COEFF = ', I8)
      STOP
      END
```

لاحظ أن استخدام البرنامج الفرعى يجعل البرنامج الأساسى سهلا وواضحاً . من ناحية أخرى ، إذا حسبت $n!$ و $(n-i)!$ و $i!$ مفصلة فى البرنامج الأساسى ، سيكون البرنامج طويلا ، وأقل وضوحاً وأكثر تعرضاً لخطأ . لاحظ أيضاً أن IBINQ يمكن أن تحسب بجملة واحدة :

```
IBINQ = IFACT(N)/(IFACT(N - I)*IFACT(I))
```

ومع ذلك ، فالتعبيرات الأطول والأكثر تعقيداً تكون أكثر تعرضاً للخطأ ، ومن الحكمة دائماً أن نجزئ التعبيرات المعقدة إلى عدة تعبيرات بسيطة .

ملاحظة : أهملنا فى الواقع احتمال حدوث فيض مع $n!$ عندما تكون قيم n كبيرة ، وسنناقش فيما بعد طرق أخرى لحساب $\binom{n}{i}$.

(ب) افرض أنه تم عقد ثلاثة امتحانات قصيرة وآخر نهائى لفصل به 25 طالباً . تثبت درجات هذه الامتحانات على بطاقات بجانب أرقام ID للطلبة بحيث يكون لكل طالب بطاقة واحدة . وتحسب درجة الطالب فى المنهج بحساب متوسط درجة الامتحان النهائى بإضافة أعلى درجة حصل عليها فى أحد الامتحانات الثلاثة القصيرة .

يحسب البرنامج التالى والذى يستخدم البرنامج الفرعى BIG درجة كل طالب .

```
       WRITE(6, 10)
  10   FORMAT('1', 4X, 'ID', 6X, 'SCORE 1', 2X, 'SCORE 2',
     1         2X, 'SCORE 3', 6X, 'FINAL', 5X, 'GRADE')
       DO 500 I = 1, 25
           READ(5, 20) ID, T1, T2, T3, FINAL
           GRADE = 0.5*(FINAL + BIG(T1, T2, T3))
           WRITE(6, 30) ID, T1, T2, T3, FINAL, GRADE
 500   CONTINUE
  20   FORMAT(I10, 4(F6.2, 4X))
  30   FORMAT(1X, I6, 5(3X, F6.2))
       STOP
       END
```

ملاحظة : يمكن أن يستدعى أى برنامج فرعى برنامجاً فرعياً آخر ، ولكن لا يمكن أن يستدعى نفسه . وبصورة أعم لا يمكن أن تشكل البرامج الداعية دائرة . فعلى سبيل المثال لا يمكن أن نتعرض للحالة التالية :

البرنامج الفرعى A يستدعى البرنامج الفرعى B

البرنامج الفرعى B يستدعى البرنامج الفرعى C

البرنامج الفرعى C يستدعى البرنامج الفرعى D

البرنامج الفرعى D يستدعى البرنامج الفرعى A

وحتى نضمن عدم حدوث ذلك يجب أن نضع البرنامج الفرعى قبل أى برنامج آخر يستدعيه ، مثلا إذا كان البرنامجان الفرعيان X و Y يستدعيان البرنامج الفرعى Z فنضع Z بعد X و Y فى مجموعة بطاقات برامج فورتران .

٧ — ٤ دوال البرامج الفرعية FUNCTION الحاسبة لعدة قيم

إلى الآن ، كل برامجنا الفرعية FUNCTION كانت تحسب قيمة واحدة ، وكانت تخصص هذه القيمة لإسم الدالة NAME في البرنامج الفرعي . سؤال : هل يستطيع برنامج فرعي FUNCTION أن يحسب عدة قيم ؟ الرد هو « نعم » (رغم أن البرنامج الفرعي SUBROUTINE والذي سيناقش في قسم ٧ — ٧ هو الذي يستخدم عادة في مثل هذه الحالة) .

تذكر أن وسيلة الاتصال الوحيدة بين البرنامج الداعي والبرنامج الفرعي تكون من خلال الخلاصات والمعاملات . وبالتالي ، كلما استخدمنا برنامجاً فرعياً FUNCTION لحساب عدة قيم ، تخصص قيمة واحدة لإسم الدالة NAME والقيم الأخرى يجب أن تخصص إلى معاملات زائفة (وبالتالي تنقل أيضاً إلى البرنامج الداعي) .

مثال ٧ — ٢

افرض أننا نريد أن نكتب البرنامج الفرعي FUNCTION الذي يحسب أكبر رقم ومجموع أي ثلاثة أرقام A و B و C . يسمى البرنامج BBB حيث تستخدم BBB لتخزين القيمة الكبرى . سيكون البرنامج الفرعي مشابهاً للبرنامج الفرعي BIG في شكل ٧ — ٢ فيما عدا أنه يجب أن يكون لـ BBB معامل رابع وليكن SUM الذي سوف يستخدم لنقل مجموع A و B و C إلى البرنامج الداعي . ويظهر مثل هذا البرنامج BBB كالتالي :

```
FUNCTION BBB(A, B, C, SUM)
SUM = A + B + C
BBB = A
IF(BBB.LT.B) BBB = B
IF(BBB.LT.C) BBB = C
RETURN
END
```

مع فرض أن البرنامج الداعي به الجملة التالية :

```
RESULT = BBB(X, Y, Z, TOTAL)
```

بعد تنفيذ هذه الجملة ، ستحتوى RESULT على أكبر الأرقام X و Y و Z وستحتوى TOTAL على مجموعها . ستكون TOTAL بعد ذلك متاحة لحسابات . على سبيل المثال ، إذا نفذت فيما بعد الجملة التالية :

```
AVE = TOTAL/3.0
```

فسوف تحتوى AVE على متوسط X و Y و Z

٧ — ٥ مجموعات متراصة وبرامج فرعية FUNCTION ، أبعاد متغيرة

يمكن أيضاً أن تستخدم مجموعة متراصة كمعامل لبرنامج فرعي FUNCTION . في مثل هذه الحالة ، يجب أن تكون الخلاصة المناظرة في جملة الاستدعاء مجموعة متراصة أيضاً . ومع ذلك يجب تعريف المجموعة المتراصة في البرنامج الفرعي بجملة DIMENSION وذلك في البرنامج الفرعي . وعلاوة على ذلك فلا يمكن أن تتجاوز أبعاده أبعاد الخلاصة المناظرة .

شكل ٧ — ٤ (أ) عبارة عن برنامج فرعي BIGG يحدد أكبر عنصر في المجموعة المتراصة الخطية A التي بها 25 عنصرا ، وشكل ٧ — ٤ (ب) عبارة عن برنامج استدعاء عملي .

سيحدد هذا البرنامج الفرعي BIGG العنصر الأكبر في مجموعة متراصة خطية بها بالتحديد 25 عنصراً ، وهذا من البديهي يحد من استخدامها . يسمح الفورتران بمرونة أكبر وذلك باستخدام الأبعاد المتغيرة (أي ، الأبعاد القابلة للتعديل) وذلك في البرامج الفرعية فقط . وسنناقش الآن هذه الخاصية .

```
      FUNCTION BIGG(A)
      DIMENSION A(25)
      BIGG = A(1)
      DO 10 K = 2, 25
           IF(BIGG.LT.A(K)) BIGG = A(K)
   10 CONTINUE
      RETURN
      END
```

(ا)

```
      DIMENSION X(25)
      REAL LARGE
      READ(5, 10) X
   10 FORMAT(5(F8.2, 2X))
      LARGE = BIGG(X)
      WRITE(6, 20) LARGE
   20 FORMAT('0', 'LARGEST VALUE IS', 2X, F8.2)
      STOP
      END
```

(ب)

شكل ٧ – ٤

الأبعاد المتغيرة . نتذكر أنه فى أى برنامج يجب أن يكون دليل اسم المجموعة المتراصة فى جملة DIMENSION ثابتاً صحيحاً ولا يمكن أن يكون متغيراً . ويكون هذا غير صحيح تماماً إذا ظهرت جملة DIMENSION فى برنامج فرعى . بالتحديد افرض أن جملة DIMENSION فى برنامج فرعى ن أن المتغير A مجموعة متراصة و A هو معامل البرنامج الفرعى . إذن يمكن أن يكون دليل A فى جملة DIMENSION متغيراً صحيحاً بشرط أن يكون المتغير أيضاً معاملا فى البرنامج الفرعى . ويتم توضيح ذلك فى البرنامج الفرعى FUNCTION التالى الذى يحدد أكبر عنصر فى متجه A عدد عناصره N .

```
      FUNCTION BIGMM(A, N)
      DIMENSION A(N)
      BIGMM = A(1)
      DO 10 K = 1, N
           IF(BIGMM.LT.A(K)) BIGMM = A(K)
   10 CONTINUE
      RETURN
      END
```

ويجدر أن نؤكد أن استخدام الأبعاد المتغيرة مسموح به فقط فى البرامج الفرعية ، أى ، لا يمكن استخدامها بأى حال من الأحوال فى البرنامج الأساسى .

ملاحظة : لا يجب أن تتجاوز قيمة المعامل N المستخدم فى جملة DIMENSION المتغيرة فى برنامج فرعى حجم البعد الأصلى للمجموعة المتراصة المناظر فى البرنامج الداعى .

مثال ٧ – ٣

(ا) اكتب البرنامج الفرعى FUNCTION الذى يحسب مجموع العناصر فى مجموعة متراصة خطية بها N عنصراً . هنا A و N معاملات :

```
      FUNCTION SUM(A, N)
      DIMENSION A(N)
      SUM = 0.0
      DO 99 K = 1, N
           SUM = SUM + A(K)
   99 CONTINUE
      RETURN
      END
```

(ب) افرض أن X و Y مجموعات متراصة خطية مخزنة فى الذاكرة ومعرفة كالآتى

```
DIMENSION X(100), Y(200)
```

استخدام التعريف السابق للدالة SUM لإيجاد المتوسطات التالية :

$(X_1 + X_2 + \cdots + X_{25})/25$

$(Y_1 + Y_2 + \cdots + Y_M)/M$

فيصبح لدينا :

```
AVEX = SUM(X, 25)/25.0
AVEY = SUM(Y, M)/FLOAT(M)
```

(افترضنا أن $M \leq 200$)

٧ — ٦ دوال الجملة الحسابية

افرض أننا نريد أن نحسب قيمة الدالة التربيعة التالية :

$$g(x) = x^2 - 5x + 2$$

لقيم X حيث : $x = 1, 2, \ldots, 20$. باستخدام برنامج FUNCTION يصبح لدينا :

البرنامج الداعى

```
     DO 100 J = 1, 20
            VALUE = G(FLOAT(J))
            WRITE(6, 10) J, VALUE
100  CONTINUE
 10  FORMAT(1X, I3, 3X, F8.2)
     STOP
     END
```

البرنامج الفرعى

```
FUNCTION G(X)
G = X*X - 5.0*X + 2.0
RETURN
END
```

لاحظ أن هذا البرنامج الفرعى FUNCTION يتكون من جملة حسابية واحدة (إلى جانب جملة التعريف FUNCTION وجملتا RETURN و END) يمكن أن نبسط مثل هذا النوع من البرامج الفرعية FUNCTION ببساطة بكتابة الجملة

```
G(X) = X*X - 5.0*X + 2.0
```

فى بداية البرنامج الداعى كما يلى :

```
C
C       DEFINE THE FUNCTION G
C
        G(X) = X*X - 5.0*X + 2.0
C
C       COMPUTE FUNCTIONAL VALUES
C
        DO 100 J = 1, 20
               VALUE = G(FLOAT(J))
               WRITE(6, 10) J, VALUE
   100  CONTINUE
    10  FORMAT(1X, I3, 3X, F8.2)
        STOP
        END
```

وتسمى الدالة G دالة الجملة الحسابية .

وفيما يلي قواعد دوال الجملة الحسابية :

١ – تعرف دالة الجملة باستخدام الشكل التالي :

تعبير حسابي = إسم (معامل 1 ، ومعامل 2 ، ... ، ومعامل n)

يجب أن تكون المعاملات متغيرات بدون أدلة ، ولا يجب أن يحتوى التعبير الحسابي على أى متغيرات بدليل . ومع ذلك، يمكن أن يحتوى التعبير الحسابي على دوال مكتبية وعلى برامج فرعية FUNCTION وعلى دوال جمل أخرى (بشرط أن تكون قد سبق تعريفها) . رغم أن بعض المترجمات يسمح بظهور جملة التعريف فى أى مكان فى البرنامج بشرط أن يكون سابقاً لاستخدامها لكن بعض المترجمات الأخرى تتطلب أن تكون موضوعة فى بداية البرنامج بعد أى جمل نوع أو تعريف ، ولكن قبل أى جملة قابلة التنفيذ .

٢ – تتبع تسمية الدالة ومعاملاتها نفس قواعد البرامج الفرعية FUNCTION والمعاملات هى ، كما فى البرامج الفرعية FUNCTION متغيرات زائفة ، ولا يتم تخصيص اماكن تخزين لها فى الذاكرة وبذلك تكون خاصة بالجملة .

٣ – قواعد استدعاء دوال الجمل هى نفس القواعد المستخدمة للبرامج الفرعية FUNCTION . أى فى أى مكان من البرنامج يتطلب فيه قيمة الدالة يكتب ببساطة اسم دالة الجملة بخلاصات مناسبة . ويمكن أن تكون الخلاصات أى تعبيرات حسابية طالما أنها متفقة بطريقة واحد إلى واحد فى الترتيب ونوع المعاملات مع جملة التعريف . على سبيل المثال ، باعتبار دالة الجملة السابقة G(X) قد عرفت فى البرنامج ، يمكن أن أى تستدعى كما يلى :

AVE = (G(A + SQRT(B)) + DEP)/2.0

حيث AVE هى متوسط DEP و G تحسب قيمتها عند A + SQRT(B) . الخلاصة هنا هى (A + SQRT(B)) وهى تقابل المعامل X فى جملة الدالة .

٤ – نقطة هامة هى أن دالة الجملة داخلية فى البرنامج (أو البرنامج الفرعى) التى تظهر فيه ، من ثم ، لا يمكن أن تستدعى بأى برنامج آخر .

حيث أن دالة الجملة داخلية لبرنامجها فلها درجة حرية ليست لدى البرامج الفرعية FUNCTION . بالتحديد ، المتغيرات التى لا تظهر فى قائمة المعاملات يمكن أن تظهر فى الجملة . على سبيل المثال ، دالة الجملة الحسابية الآتية صحيحة لغوياً :

F(X, Y) = A*X**2 + B*X*Y + C*Y**2

فى هذه الحالة X و Y متغيرات زائفة ، ومن ثم ، يمكن أن تستخدم X و Y كأسماء متغيرات فى مكان آخر فى البرنامج .

ومع ذلك فإن A و B و C ليست متغيرات زائفة حيث أنها ليست فى قائمة المعاملات ، ولذلك سوف نفترض أنه تم تعريفها عند نداء الدالة . على سبيل المثال :

F(1.0, 2.0)

ستحسب الدالة لقيم A و B و C الحالية .

مثال ٧ – ٤

نفرض مجموعة N من البطاقات ، كل بطاقة مثقب عليها ثلاثة أرقام حقيقية ، هي المعاملات A و B و C لمعادلة من الدرجة الثانية : .

$$F(X) = AX^2 + BX + C$$

اكتب برنامج فورتران لحساب قيمة الدالة عندما $X = -5, -4, \ldots, 5$ لكل ثلاثى A و B و C .

. لاحظ أن هناك 11 قيمة لـ X وسنستخدم حلقة DO بدليل $I = 1, 2, \ldots, 11$ والعلاقة بين المعامل I وقيم X يمكن أن تعرض كتالى :

I:	1,	2,	3,	4,	5,	6,	7,	8,	9,	10,	11
X:	−5,	−4,	−3,	−2,	−1,	0,	1,	2,	3,	4,	5

لاحظ أن X و I تربطهما العلاقة $I - X = 6$ من ثم :

$$X = -6 + I$$

فيما يلى البرنامج المطلوب :

```
C
C       EXAMPLE USING STATEMENT FUNCTION
C
        F(X) = A*X*X + B*X + C
        READ(5, 5) N
      5 FORMAT(I3)
        DO 100 J = 1, N
                READ(5, 10) A, B, C
     10         FORMAT(3F5.1)
                WRITE(6, 15) A, B, C
     15         FORMAT('1', 'A = ', F5.1/1X, 'B = ', F5.1/1X, 'C = ', F5.1)
C
C               EVALUATING THE FUNCTION
C
                DO 200 I = 1, 11
                        Y = F(-6.0 + FLOAT(I))
                        WRITE(6, 20) I, Y
    200         CONTINUE
     20         FORMAT(1X, 3I, 2X, F8.2)
    100 CONTINUE
        STOP
        END
```

٧ – ٧ برامج صغيرة فرعية SUBROUTINES

هناك نوع آخر من البرامج الفرعية يسمى برنامجاً صغيراً فرعياً SUBROUTINE . تذكر ما سبق شرحه أن اسم البرنامج الفرعى FUNCTION تخصص له دائما قيمة فى البرنامج الفرعى . هذا ليس صحيحاً بالنسبة للبرامج الصغيرة الفرعية SUBROUTINE . وما عدا هذا الفرق وما يترتب عليه ، تتشابه القواعد والقيود المتحكمة فى كتابة البرامج الصغيرة الفرعية SUROUTINE مع تلك الخاصة بالبرامج الفرعية FUNCTION يجب أن نلاحظ أننا نستخدم SUBROUTINE

أكثر من البرنامج الفرعى FUNCTION عندما يكون لدينا قيم متعددة نريد حسابها . علاوة على حساب قيم متعددة ، نستخدم البرامج الصغيرة الفرعية SUBROUTINE لإنجاز مهام (مثل استبدال عناصر .. إلخ) .

كما هو الحال فى البرامج الفرعية FUNCTION يجب الإعلان أولا عن البرامج الصغيرة الفرعية SUBROUTINE باستخدام جملة التعريف SUBROUTINE ويأخذ الإعلان عن البرنامج الفرعى الصورة العامة التالية :

SUBROUTINE NAME(param1, param2, . . . , param*n*)

وحيث أن اسم البرنامج الفرعى الصغير (NAME) لا تخصص له قيمة فى البرنامج الفرعى فلا داعى لمناقشة موضوع النوع هنا . (ومع ذلك لا يمكن أن نتجاوز أسماء البرامج الفرعية الصغيرة ستة حروف) .

وهو كأى برنامج فرعى (Subprogram) ، فإن SUBROUTINE أيضاً برنامج كامل ومستقل . من ثم ، يمكن أن يحتوى على كل خصائص أى برنامج : إعلانات النوع وجمل DIMENSION وبرامج فرعية FUNCTION أو وبرامج فرعية صغيرة SUBROUTINE وجملة END وهكذا . وعلاوة على ذلك فالاتصال بين البرنامج الداعى والبرامج الصغير الفرعى SUBROUTINE يكون فقط من خلال المعاملات ، ومن ثم فأسماء المتغيرات (فضلا عن المعاملات) و أرقام الجمل فى SUBROUTINE تعتبر محلية (داخلية) لهذا البرنامج الفرعى . وسنناقش فى جزء متأخر من هذا القسم طريقة استدعاء SUBROUTINE وطريقة نقل القيم من خلال المعاملات) .

نعطى الآن مثالين للبرامج الصغيرة الفرعية SUBROUTINE سيؤدى الأول مهمة وسيحسب الثانى قيما .

١ – بفرض أن X و Y قد سبق تعريفهما يستبدل فجزء البرنامج التالى القيم فى X و Y

```
T = X
X = Y
Y = T
```

نستطيع أن نكتب برنامجاً صغيراً فرعياً SUBROUTINE ويسمى INTCHG لتأدية عملية التبديل :

```
SUBROUTINE INTCHG(X, Y)
T = X
X = Y
Y = T
RETURN
END
```

لاحظ أولا أن الاسم INTCHG لا تخصص له قيمة فى البرنامج الصغير الفرعى (وعلى ذلك فبدء الاسم بالحرف I لا يهم) بفرض أن القيم المستبدلة سوف تستخدم فيما بعد فى البرنامج الداعى لدينا جملة RETURN التى تنقل التحكم ثانية إلى البرنامج الداعى .

٢ – ادرس مرة أخرى مسألة كتابة البرنامج الفرعى الذى يحسب القيمة الكبرى ومجموع ثلاثة أرقام A و B و C يمكن عمل هذا بالبرنامج الصغير الفرعى SUBROUTINE حيث نستخدم الخلاصات لنقل القيمة المحسوبة ثانية إلى البرنامج الداعى . فيما يلى مثال لهذا البرنامج الفرعى SUBROUTINE :

```
SUBROUTINE LARGE(A, B, C, BIG, SUM)
SUM = A + B + C
BIG = A
IF(BIG.LT.B) BIG = B
IF(BIG.LT.C) BIG = C
RETURN
END
```

لاحظ التشابه والفرق بين البرنامج الفرعي هذا وبين البرنامج الفرعي FUNCTION فى مثال ٧ – ٢ . الاسم LARGE هنا ليس له قيمة ، ومن ثم ، BIG الأكبر بين A و B و C يجب أن تذكر كعامل حتى يمكن نقلها ثانية إلى البرنامج الداعى .

كيف نستدعى البرامج الصغيرة الفرعية SUBROUTINE ؟ كما سبق أن ذكرنا هناك اختلافاً عن البرنامج الفرعى FUNCTION فاسم البرنامج الفرعى SUBROUTINE ليست له قيمة ، وعلى ذلك فطلوب نوع جديدمن الجمل – جملة CALL فجملة النداء التالية تستدعى البرنامج الصغير الفرعى INTCHG

CALL INTCHG(A(3), T)

والتأثير النهائى لهذه الجملة فى البرنامج الداعى هو استبدال القيم A(3) و T

نلاحظ أنه بعد استدعاء INTCHG تنقل عناوين A(3) و T إلى البرنامج الفرعى ويعوض بهما فيه بدلا عن المتغيرات الزائفة X و Y . حقيقة أن الاسم T فى البرنامج الداعى مستخدمة أيضاً كاسم فى البرنامج الفرعى INTCHG وذلك لا يسبب أى تشويش لأن المتغير T فى SUBROUTINE داخل لهذا البرنامج الفرعى ويستخدم فى تنفيذ SUBROUTINE عنوان الخلاصة الداعية T (وليس الاسم T) .

جملة النداء التالية صحيحة لغوياً أيضاً :

CALL LARGE(U + V, W – SQRT(V), U*U, T, S)

التأثير النهائى لهذه الجملة فى البرنامج الداعى هو أن S ستحتوى على المجموع U + V و W — SQRT(V) و U*U وستحتوى T على أكبر قيمة من القيم الثلاثة .

أننا نؤكد مرة ثانية ، كما فى حالة البرامج الفرعية FUNCTION أنه يجب أن تتفق الخلاصات بطريقة واحد – إلى – واحد فى جملة النداء مع ترتيب ونوع المعاملات فى جملة التعريف SUBROUTINE نوضح هذه الحقيقة باستخدام جملة CALL السابقة :

(أ) قائمة الخلاصات	U + V,	W – SQRT(V),	U*U,	T,	S
	↓	↓	↓	↓	↓
(ب) قائمة المعاملات	A,	B,	C,	BIG,	SUM

مثال ٧ – ٥

افرض أن A مجموعة متراصة تحتوى على N عنصر على الأكثر . اكتب البرنامج الصغير الفرعى SUBROUTINE الذى :

١ – يحدد أكبر قيمة من عدد العناصر K الأولى :

A(1), A(2), . . . , A(K).

٢ – يحسب مجموع عدد العناصر K الأولى .

A(1) + A(2) + · · · + A(K)

ينجز البرنامج الصغير الفرعي SEEK ما سبق التوصل إليه :

```
      SUBROUTINE SEEK(A, N, K, HIGH, SUM)
      DIMENSION A(N)
      SUM = 0.0
      HIGH = A(1)
      DO 100 J = 1, K
            SUM = SUM + A(J)
            IF(HIGH.LT.A(J)) HIGH = A(J)
  100 CONTINUE
      RETURN
      END
```

كما نرى فى المثال السابق ، يمكن استخدام أسماء مجموعات متراصة كعاملات فى برنامج فرعى SUBROUTINE (بشرط أن تظهر أيضاً فى جملة DIMENSION فى البرنامج الفرعى) ، ويمكن أن تأخذ أبعاداً متغيرة . أى تتشابه القواعد المتحكمة فى المجموعات المتراصة فى البرامج الفرعية SUBROUTINE مع تلك الخاصة بالبرامج الفرعية FUNCTION :

مثال ٧ — ٦

افرض أن A مجموعة متراصة بها N عنصر مخزنة فى ترتيب تصاعدى ، وبالتحديد $A(1) \le A(2) \ldots \le A(N)$. اكتب برنامجاً صغيراً فرعياً SUBROUTINE ليعيد ترتيب عناصر المجموعة المتراصة فى ترتيب تنازلى ، أى $A(1) \ge A(2) \ldots \ge A(N)$. يمكننا بكل تأكيد فرز المجموعة المتراصة فى ترتيب تنازلى ، إلا أن هذه العملية ستكون كفاءتها منخفضة لأن A كما نعلم — مرتبة ترتيباً تصاعدياً . وأبسط الطرق لتنفيذ هذه المهمة هى قلب المجموعة المتراصة ، أى نستبدل A(1) و A(N) ونستبدل A(2) و A(N — 1) وهكذا . ونطلق على هذا البرنامج الصغير الفرعى INVERT وهو يستدعى البرنامج الصغير الفرعى INTCHG :

```
      SUBROUTINE INVERT(A, N)
      DIMENSION A(N)
      NN = N/2
      K = N + 1
      DO 20 I = 1, NN
            CALL INTCHG(A(I), A(K – I))
   20 CONTINUE
      RETURN
      END
```

(لاحظ جملة DO هل يستطيع القارئ أن يقول السبب فى أن حلقة DO تستمر إلى N/2 فقط بدلا من أن تستمر إلى N ؟)

يقطع استخدام البرامج الفرعية المشكلة الكبيرة والمعقدة إلى وحدات برامج أصغر . تكون كل وحدة صغيرة أسهل فى التنفيذ واكتشاف وتصحيح الأخطاء . علاوة على ذلك ، يمكن أيضاً تنفيذ وحدات البرامج الصغيرة بواسطة عدد من أفراد الفريق . وبالتالى ، يمكن تنفيذ عملية كتابة برنامج كبير على التوازى بدلا من أن تتم على التوالى .

٧ — ٨ مقارنة SUBROUTINE مع FUNCTION

تستخدم البرامج الفرعية FUNCTION بصفة عامة ، لحساب قيمة واحدة ، فى حين تستخدم البرامج الفرعية SUBROUTINE لحساب عدة قيم أو تنفيذ مهام محددة (مثل استبدال قيمتين) . نلاحظ أى شئ ينفذ ببرنامج فرعى FUNCTION يمكن أن يتم ببرنامج فرعى SUBROUTINE . وبالعكس ايضا معظم المهام المنفذة ببرامج فرعية SUBROUTINE يمكن تنفيذها ببرامج فرعية FUNCTION على سبيل المثال ، نجد أن البرنامج الفرعى FUNCTION لتبديل قيم X و T كما يلى :

```
FUNCTION EXCHG(X, Y)
T = X
X = Y
Y = T
EXCHG = 0.0
RETURN
END
```

لاحظ أن اسم الـ FUNCTION يخصص له قيمة اختيارية نظراً لأنه يجب أن يعرف فى البرنامج الفرعى . وحيث أن الاتصال يتم عن طريق المعاملات ، فإن قيم X و Y ستتبدل فى البرنامج الأساسى . إلا أن هذه التركيبة صناعية نوعاً ما وغير طبيعية . وفيما يلى الفروق الأساسية بين البرامج الفرعية SUBROUTINE و FUNCTION :

١ - لا تخصص قيمة لاسم البرنامج الفرعى الصغير SUBROUTINE (NAME) ، وحيث أنه يجب أن تكون هناك قيمة عددية أو منطقية لاسم البرنامج الفرعى FUNCTION ، من ثم ، يجب أن ينطبق على اسم (NAME) البرنامج الفرعى FUNCTION مفهوم النوع ، ويجب أن يعرف فى البرنامج الفرعى .

٢ - يمكن أن يستدعى SUBROUTINE فقط بواسطة جملة نداء خاصة وهى جملة CALL . يجب أن يستخدم اسم البرنامج الفرعى FUNCTION بنفس طريقة الدوال المكتبية . أى ، فى تعبيرات حسابية ، إلخ .

٣ - يجب أن يكون البرنامج الفرعى FUNCTION على الأقل خلاصة واحدة ، بينما يمكن ألا تكون هناك أى خلاصات للبرنامج الفرعى الصغير SUBROUTINE (انظر مسألة ٧ - ١١) .

٤ - حيث أن البرنامج الفرعى FUNCTION يحسب على الأقل قيمة واحدة يجب أن يحتوى على جملة RETURN أما البرنامج الفرعى الصغير SUBROUTINE فيمكن ألا يحتوى على جملة RETURN (انظر مسألة ٧ - ١٢) .

مسـائل محلولة

برامج فرعية صغيرة

٧ - ١ اكتشف الأخطاء ، إن وجدت ، فى كل من جمل تعريف البرامج الفرعية الصغيرة الآتية :

(ا) FUNCTION, NEXT(A, B + C, X) (جـ) SUBROUTINE NEW(X, Y(3), Z)

(ب) FUNCTION GRADE A, B, C (د) SUBROUTINE(U, V, W)

(ا) لا يجب أن تكون هناك فصلة بعد FUNCTION . أيضاً لا يمكن أن تكون B + C معامل .

(ب) يجب أن تحاط المعاملات A و B و C بأقواس .

(جـ) لا يمكن أن يكون المتغير ذو الدليل Y(3) معامل فى جملة تعريف برنامج فرعى .

(د) البرنامج الفرعى الصغير ليس له اسم .

٧ - ٢ اكتشف الأخطاء ، إن وجدت ، فى كل جزء من الأجزاء التالية :

```
(ا) FUNCTION XXX(A, N, K)
    DIMENSION A(N, N + 1), B(K)

(ب) SUBROUTINE YYY(A, N, K)
    DIMENSION A(J, K), B(N)
```

(ا) لا يمكن أن يظهر التعبير الحسابى N + 1 فى جملة DIMENSION

(ب) لا يمكن أن يظهر المتغير J فى جملة DIMENSION حيث أنه لا يظهر كعامل فى جملة التعريف .

٧ - ٣ اكتشف الأخطاء ، إن وجدت ، فى كل برنامج فرعى مما يلى :

```
(ا) FUNCTION ADD(X, Y, Z)
    Z = X + Y
    RETURN
    END

(ب) SUBROUTINE AAA(X, Y, Z)
    Z = X + Y
    RETURN
    END
```

(ا) يجب تخصيص قيمة لاسم الدالة ADD فى البرنامج الفرعى ، ولم يحدث هذا .

(ب) لا توجد أخطاء (لاحظ عدم تخصيص قيمة لاسم البرنامج) .

٧ - ٤ عرف دالة الجملة لحساب $R = \sqrt{u^2 + v^2 + w^2}$: واستخدمها لحساب قيمة

$$A = \frac{x}{\sqrt{x^2 + y^2 + z^2}}, \qquad B = \sqrt{x^4 + y^4 + z^4}, \qquad C = \sqrt{4x^2 + 9y^2 + 4z^2}$$

نلاحظ أن $x^4 = (x^2)^2$ و $4x^2 = (2x)^2$ وهكذا .

```
R(U, V, W) = SQRT(U*U + V*V + W*W)
A = X/R(X, Y, Z)
B = R(X*X, Y*Y, Z*Z)
C = R(2.0*X, 3.0*Y, 2.0*Z)
```

٧ - ٥ أوجد المخرجات للبرامج التالية والتى تستخدم دالة الجملة :

```
(ا)    JF(M) = M**2 - 3*M + 4
       K = 2
       L = JF(K + 2)
       M = JF(L - 3*K) + K
       WRITE(6, 10) K, L, M
10     FORMAT(1X, 3(I10, 2X))
       STOP
       END

(ب)    F(X, Y) = A*X + B*Y
       X = 2.0
       Y = 3.0
       A = 4.0
       B = 5.0
       C = F(6.0, 7.0)
       WRITE(6, 10) C
10     FORMAT(1X, F10.2)
       STOP
       END
```

(ا) يعرف السطر الأول الدالة $JF(M) = M^2 - 3M + 4$ ويخصص السطر الثانى 2 إلى K . وتحسب قيمة L و M بواسطة السطرين الثالث والرابع كما يلى :

$$L \leftarrow JF(2 + 2) = JF(4) = 4^2 - 3\cdot4 + 4 = 8$$
$$M \leftarrow JF(8 - 3\cdot2) + 2 = JF(2) + 2 = (2^2 - 3\cdot2 + 4) + 2 = 4$$

ومن ثم ستطبع 2 و 8 و 4 على سطر واحد ، كل فى حقل بعرض 10 .

(ب) تخصص 2 و 3 و 4 و 5 إلى X و Y و A و B على الترتيب وذلك بواسطة الأسطر من الثانى إلى الخامس . تذكر أن X و Y فى F(X, Y) متغيرات زائفة وليس لهما علاقة بالقيم X و Y فى البرنامج . بالتحديد ، عند حساب قيمة C بواسطة السطر السادس وباستخدام قيم A و B فى البرنامج ، نعوض عن X بالرقم 6 وعن Y بالرقم 7 فى F(X, Y) وبذلك يصبح :

$$C \leftarrow 4 \cdot 6 + 5 \cdot 7 = 59$$

وبالتالى ، ستطبع 59.00 مضبطة من الطرف الأيمن فى أول 10 أعمدة من صفحة الطباعة .

٧ – ٦ افرض أننا أعطينا البرامج الفرعية الآتية :

```
FUNCTION SUM(X, Y, Z)
SUM = X + Y + Z
Y = X + Y
Z = Y + Z
X = Z + X
RETURN
END
```

```
SUBROUTINE ADD(X, Y, Z, TOTAL)
Y = X + Y
Z = Y + Z
X = Z + X
TOTAL = X + Y + Z
RETURN
END
```

أوجد الخرج لجزء البرنامج التالى :

```
   A = 1.0
   B = 2.0
   C = 3.0
   AMOUNT = SUM(A, B, C)
   CALL ADD(A, B, C, GT)
   WRITE(6, 10) A, B, C, AMOUNT, GT
10 FORMAT(1X, 5(F4.1, 2X))
```

بعد تنفيذ السطر الرابع يصبح لدينا A = 7.0 و B = 3.0 و C = 6.0 و AMOUNT = 6.0 . وبعد تنفيذ السطر الخامس يصبح لدينا A = 23.0 و B = 10.0 و C=16.0 و GT = 49.9 ومن ثم يتكون الخرج من 23.0 و 10.0 و 16.0 و 6.0 و 49.0

برامج

٧ – ٧ اكتب البرنامج الفرعى SUBROUTINE الذى يحسب الرصيد الجديد لحساب مراجعة ، ليكن فى الشكل :

BANK(BAL, DEP, M, CK, N)

حيث :

BAL تدل على الرصيد الشهرى لحساب المراجعة

و DEP مجموعة متراصة تعطى قائمة M بودائع الشهر .

و CK مجموعة متراصة تعطى قائمة بالمدفوعات على عدد N من الشيكات خلال الشهر .

افترض أن هناك مقابل خدمة (SC) عبارة عن $1.00 شهرياً و 5¢ لكل شيك و 2¢ لكل وديعة . أيضاً اعتبر أن DEP و CK لا تتجاوز أبدا 100 عنصر.

نجمع ببساطة الودائع ونطرح الشيكات ومقابل الخدمة . ومع ذلك يجب أيضاً أن نأخذ فى الاعتبار الحالة التى تكون فيها M أو N أصفار . وفيما يلى البرنامج الفرعى :

```
      SUBROUTINE BANK(BAL, DEP, M, CK, N)
      DIMENSION DEP(100), CK(100)
      IF(M.EQ.0) GO TO 10
      DO 100 K = 1, M
         BAL = BAL + DEP(K)
100   CONTINUE
 10   IF(N.EQ.0) GO TO 20
      DO 200 K = 1, N
         BAL = BAL - CK(K)
200   CONTINUE
 20   SC = 1.00 + M*0.02 + N*0.05
      BAL = BAL - SC
      RETURN
      END
```

٧ – ٨ اكتب البرنامج الفرعى SUBROUTINE الذى ليس له أى (١) خلاصة (ب) جملة RETURN

(١) فيما يلى برنامج فرعى SUBROUTINE بدون أى خلاصة .

```
      SUBROUTINE NEW
      WRITE(6, 10)
10    FORMAT(1X, 'NEW DEPOSITOR')
      RETURN
      END
```

وتكون جملة الاستدعاء من البرنامج كما يلى :

```
CALL NEW
```

ويكون لها تأثير طباعة الرسالة التالية :

```
NEW DEPOSITOR
```

(ب) لا يحتاج البرنامج الفرعى SUBROUTINE أن يرجع إلى البرنامج الداعى . بالتحديد ، لو وضعنا فى مكان جملة RETURN فى أى برنامج فرعى SUBROUTINE جملة STOP سينتهى تنفيذ برنامج الفورتران مع SUBROUTINE على سبيل المثال ، لو وضعنا STOP مكان RETURN فى (١) سيتوقف البرنامج بعد طباعة NEW DEPOSITOR

٧ – ٩ افرض أن A و B و C و D مجموعات متراصة خطية مخزنة فى الذاكرة لها 100 و 50 و 75 و 200 عنصر على الترتيب . افرض L و K مخزنة أيضا فى الذاكرة $5 \le L \le 75$ و $L < K \le 200$ اكتب جزء برنامج فورتران لحساب ما يلى :

1. $A_{15} + A_{17} + A_{19} + \cdots + A_{77}$
2. $B_{22} + B_{23} + B_{24} + \cdots + B_{36}$
3. $C_2 + C_4 + C_6 + \cdots + C_L$
4. $D_L + D_{L+1} + D_{L+2} + \cdots + D_K$

لاحظ أن كلا من الحسابات تشتمل على إيجاد مجموع حدود فى مجموعة متراصة خطية . وبدلا من عمل كل من الحسابات فى البرنامج نكتب دالة برنامج فرعى تدعى SUM لحساب $X_{IN} + X_{IN+IC} + \cdots + X_{IT}$ حيث X

مجموعة متراصة خطية بها N عنصر ، IN الدليل الابتدائى و IC معامل الزيادة و IT الدليل النهائى . وفيما يلى البرنامج الفرعى الذى ينفذ ذلك وكذا جزء برنامج الفورتران الذى يستدعيه .

البرنامج الفرعى

```
      FUNCTION SUM(X, N, IN, IT, IC)
      DIMENSION X(N)
      SUM = 0.0
      DO 20 I = IN, IT, IC
         SUM = SUM + X(I)
20    CONTINUE
      RETURN
      END
```

جزء برنامج الفورتران

```
T1 = SUM(A, 100, 15, 77, 2)
T2 = SUM(B, 50, 22, 36, 1)
T3 = SUM(C, 75, 2, L, 3)
T4 = SUM(D, 200, L, K, 1)
```

٧ – ١٠ افرض X و Y مجموعات متراصة خطية كل بها N عنصر . اكتب جزء برنامج لحساب ما يلى

$$\frac{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \cdot \sqrt{y_1^2 + y_2^2 + \cdots + y_n^2}}{\sqrt{x_1y_1 + x_2y_2 + \cdots + x_ny_n}}$$

باستخدام (FUNCTION) برنامج فرعى INNPRO الذى يحسب $a_1b_1 + a_2b_2 + \ldots. + a_nb_n$

البرنامج الفرعى

```
      REAL FUNCTION INNPRO(A, B, N)
      DIMENSION A(N), B(N)
      INNPRO = 0.0
      DO 100 K = 1, N
         INNPRO = INNPRO + A(K)*B(K)
100   CONTINUE
      RETURN
      END
```

جزء برنامج الفورتران

```
UP1 = SQRT(INNPRO(X, X, N))
UP2 = SQRT(INNPRO(Y, Y, N))
DOWN = SQRT(INNPRO(X, Y, N))
ANGLE = UP1*UP2/DOWN
```

(لاحظ إعلان النوع حيث أن INNPRO ستكون حقيقية).

٧ – ١١ اكتب برنامجاً فرعياً SUBROUTINE يطبع مجموعة متراصة ذات بعدين صفاً بصف (مع فرض أن المجموعة المتراصة لا تحتوى على أكثر من 13 عموداً) سيكون للبرنامج بعد متغير يمكن ضبطه .

```
      SUBROUTINE PRINT(A, M, N)
      DIMENSION A(M, N)
      DO 100 J = 1, M
         WRITE(6, 10) (A(J, K), K = 1, N)
10       FORMAT(1X, 13(2X, F8.2))
100   CONTINUE
      RETURN
      END
```

٧ – ١٢ افرض A مصفوفة (L و N) مخزنة فى الذاكرة . اكتب البرامج الفرعية الصغيرة SUBROUTINE التالية :

(١) SUBROUTINE FIND(A, N, L, K, J)

الذى يجد الصف J بحيث تحتوى A(J, K) على القيمة الكبرى المطلقة بين A(K, K),A(K + 1, K), ..., A(N, K) (انظر مسألة ٦ – ٩) .

(ب) SUBROUTINE CHANGE(A, N, L, K, J)

الذى يستبدل عناصر الصف K فى المصفوفة A مع العناصر المناظرة للصف J (انظر مسألة ٦ – ٩) .

(جـ) SUBROUTINE ROWMUL(A, N, L, K, J, D)

الذى يجمع قيم الصف K إلى قيم الصف J عدد D من المرات (هذه البرامج الفرعية SUBROUTINE مستخدمة فى حل المعادلات الخطية بطريقة حذف جاوس (Gauss elimination) والتى تناقش فى الفصل الثامن)

(أ)

```
      SUBROUTINE FIND(A, N, L, K, J)
      DIMENSION A(N, L)
      J = K
      KK = K + 1
      DO 100 I = KK, N
            IF(ABS(A(J, K)).LT.ABS(A(I, K))) J = I
100   CONTINUE
      RETURN
      END
```

(ب)

```
      SUBROUTINE CHANGE(A, N, L, K, J)
      DIMENSION A(N, L)
      IF(J.EQ.K) RETURN
      DO 100 I = 1, L
            T = A(K, I)
            A(K, I) = A(J, I)
            A(J, I) = T
100   CONTINUE
      RETURN
      END
```

لاحظ أننا سمحنا بإمكانية تساوى K و J ، فى هذه الحالة ، ليس علينا أن نمر خلال حلقة DO التكرارية .

(جـ)

```
      SUBROUTINE ROWMUL(A, N, L, K, J, D)
      DIMENSION A(N, L)
      DO 100 I = 1, L
            A(J, I) = A(J, I) + D*A(K, I)
100   CONTINUE
      RETURN
      END
```

مســـائل تكميلية

البرامج الفرعية

٧ – ١٣ اكتشف الأخطاء ، إن وجدت ، فى كل جملة تعريف للبرامج الفرعية التالية :

(أ) FUNCTION, ORDER(A, B, 2*X)

(ب) FUNCTION(I, J, K)

(جـ) SUBROUTINE SEEK(A, B, X, M (K), Z)

(د) SUBROUTINE LOOK(K, J(4), L, M)

٧ – ١٤ اكتشف الأخطاء ، إن وجدت ، فى كل جزء من برامج الفورتران التالية :

(أ)
```
FUNCTION BANK(A, B, N, K)
DIMENSION A(N, N − 1), C(K), B(200)
```

(ب)
```
SUBROUTINE LOOK(X, M, N, Z)
DIMENSION X(M), Y(M, K), Z(N)
```

٧ – ١٥ اكتشف الأخطاء ، إن وجدت ، فى البرامج الصغيرة الفرعية التالية :

(أ)
```
FUNCTION DEP(X, Y, N)
Z = X + FLOAT(N)
Y = X − N*Z
RETURN
END
```

(ب)
```
REAL SUBROUTINE LOOK(X, Y, Z)
Z = X + Y
RETURN
END
```

٧ – ١٦ أوجد خرج البرنامج التالى :

```
       JG(K) = K**2 + L*K + M
       K = 3
       L = 4
       M = 5
       I = JG(6)
       J = JG(L)
       WRITE(6, 10) I, J
   10  FORMAT(1X, 2I10)
       STOP
       END
```

٧ – ١٧ (أ) اكتب البرنامج الفرعى FUNCTION ADD(M, N) الذى يحسب مجموع N من الأعداد الصحيحة المتعاقبه ابتداء من M بالتحديد .

$$M + (M + 1) + (M + 2) + \cdots + (M + N - 1)$$

على سبيل المثال MM = ADD(5, 4) سيخزن الرقم : $5 + 6 + 7 + 8 = 26$ فى MM

(ب) أوجد خرج البرنامج التالى الذى يستخدم البرنامج الفرعى FUNCTION السابق :

```
C      PROGRAM USING FUNCTION SUBPROGRAM
       I = 1
       J = 2
       K = 3*I + 2*J
       I = ADD(J, K − 3)
       J = ADD(I − 1, J + 1)
       K = ADD(K, K − 3)
       WRITE(6, 10) I, J, K
   10  FORMAT(1X, 3I10)
       STOP
       END
```

٧ - ١٨ (أ) اكتب برنامجاً فرعياً صغيراً SUBROUTINE SUB (J,K, JSUM, JPROD, JDIFF) بحيث تعطى JSOM و JPROD و JDIFF تعطى المجموع J + K وحاصل الضرب J*K والفرق J — K على الترتيب.

(ب) أوجد خرج البرنامج التالى الذى يستخدم البرنامج الفرعى الصغير السابق :

```
C       PROGRAM USING SUBROUTINE SUBPROGRAM
        J = 3
        K = 2
        CALL SUB(J, K, L, M, N)
        CALL SUB(L + M, 3*N, J, N, M)
        WRITE(6, 10) J, K, L, M, N
     10 FORMAT(1X, 5(I5, 2X))
        STOP
        END
```

٧ - ١٩ اكتب برنامجاً حقيقياً FUNCTION ADDS(J, K) يحسب المجموع

$1/J + 1/(J + 1) + 1/(J + 2) + \cdots + 1/K$

عندما تكون $J \leq K$ ولكن يضع $ADDS = 0.0$ عندما تكون $J > K$. أوجد خرج البرنامج التالى :

```
   I = 2
   J = 4
   K = 3
   X = ADDS(I, J)
   Y = ADDS(J, K)
   Z = ADDS(J, 5)
   WRITE(6, 30) X, Y, Z
30 FORMAT(1X, 3F12.2)
   STOP
   END
```

٧ - ٢٠ اكتب جزء من برنامج فورتران يحساب :

$$A = x^2 + y^2 - 25$$
$$B = 4x^2 + 9y^2 - z^2$$
$$C = \sqrt{x^2 + y^2 - 36}$$

(تلميح : استخدم دالة الجملة الحسابية $F(r, s, t) = r^2 + s^2 - t^2$)

برامج

٧ - ٢١ إذا أعطيت مجموعة متراصة A $(A_1, A_2, \ldots, A_N)$ بها N عنصر ، وكان لدينا تعريف المتوسط الحسابي على أنه :

$(A_1 + A_2 + \cdots + A_N)/N$

والمتوسط الهندسى بالآتى :

$\sqrt[N]{A_1 \cdot A_2 \cdot \ldots \cdot A_N}$

فعليك أن :

(أ) تكتب برنامجاً فرعياً SUBROUTINE لحساب المتوسط الحسابى والهندسى .

(ب) تكتب برنامجاً فرعياً FUNCTION لحساب المتوسط الحسابى والهندسى .

٧ – ٢٢ افرض أن C مصفوفة مكونة من (N × N) . اكتب البرنامج الفرعى FUNCTION الذى يجد حاصل الضرب لعناصر القطر C(1, 1), C(2, 2), . . . , C(N, N)

٧ – ٢٣ افرض أن B مصفوفة مكونة من M × N :

(أ) اكتب البرنامج الفرعى FUNCTION الذى يجد أكبر عنصر فى B

(ب) اكتب البرنامج الفرعى FUNCTION الذى يجد العنصر ذو أكبر قيمة مطلقة B .

(جـ) اكتب البرنامج الفرعى FUNCTION الذى يجد مجموع عناصر B .

٧ – ٢٤ إذا أعطيت أربعة امتحانات لكل طالب فى فصل به 25 طالباً . وثقبت درجات كل طالب على بطاقة بالإضافة إلى ID الطالب أو الطالبة . وكانت درجة المقرر تحسب كمتوسط أفضل ثلاث درجات امتحان . اكتب برنامج لحساب الدرجات للطلبة (تلميح : طريقة لحساب مجموع أفضل ثلاث درجات هى طرح أقل درجة من مجموع كل الدرجات الأربع . ومن ثم ، اكتب برنامجاً فرعياً FUNCTION لإيجاد أقل درجة ثم استخدم هذا البرنامج الفرعى فى كتابة البرنامج الأساسى) .

٧ – ٢٥ اكتب البرنامج الفرعى الصغير SUBROUTINE فى الشكل :

STAT(A, N, SUM, AVE, VAR, SD)

الذى يحسب SUM والمتوسط الحسابى AVE والتباين VAR والانحراف المعيارى SD لعناصر مجموعة متراصة خطية A بها N عنصراً (انظر مسألة ٦ – ٦)

٧ – ٢٦ افرض A مجموعة متراصة خطية بها N عنصراً . اكتب البرامج الفرعية الصغيرة التالية SUBROUTINE

(أ) SORTUP(A, N) الذى يرتب A ترتيباً تصاعديا (انظر مسألة ٦ – ١٢)

(ب) SORTDN(A, N) الذى يرتب A ترتيباً تنازلياً .

٧ – ٢٧ اكتب البرنامج الفرعى الصغير SUBROUTINE فى الشكل

DEPR(COST, N, A)

حيث تتناقص COST بطريقة مجموع الحدود على عدد N من السنوات (انظر مسألة ٥ – ٣٧) ، و A مجموعة متراصة خطية بها N عنصراً بحيث تدل A(K) على الكمية المتناقصة فى السنة K .

٧ – ٢٨ افرض أن J و K عددين صحيحين موجبين . اكتب البرامج الفرعية FUNCTION التالية :

(أ) LGCD(J, K) الذى يجد القاسم المشترك الأعلى لـ J و K

(ب) LCM(J, K) الذى يجد المضاعف المشترك الأصغر J و K

تلميح : (LCM(J, K). LGCD(J, K) = J. K

٧ – ٢٩ افرض A مصفوفة (M × N) فاكتب البرامج الفرعية التالية :

(أ) MAXROW(A, M, N, K) الذى يحدد موضع أكبر عنصر فى الصف K من المصفوفة A .

(ب) MAXCOL(A, M, N, L) الذى يحدد موضع أكبر عنصر فى العمود L من المصفوفة A .

٧ - ٣٠ اجعل A مصفوفة مربعة (N × N)

(ا) اكتب البرنامج الفرعى SYM الذى يعطى القيمة M = 1 إذا كانت المصفوفة A متماثلة ويعطى القيمة M = 0 إذا كانت غير ذلك . والمصفوفة A تكون متماثلة إذا كانت $A_{ij} = A_{ji}$ لكل من i و j .

(ب) اكتب البرنامج الفرعى TRANS الذى يقبل A ثم يحورها ويعيد تخزينها فى A أيضاً (تعرف المصفوفة المحورة A بـ A^T وهى المصفوفة A (N × N) حيث $A_{ij} = A_{ji}$ لكل من i و j) .

٧ - ٣١ افرض A مصفوفة (N × N) من الممكن تخزين العناصر فى جزء المثلث العلوى لـ A داخل مجموعة متراصة خطية B بها N(N + 1)/2 عنصر كما يلى :

```
A(1, 1)   A(1, 2)    A(1, 3)    ...   A(1, N)
   ↓         ↓          ↓                ↓
 B(1)      B(2)       B(3)      ...     B(N)
          A(2, 2)    A(2, 3)    ...   A(2, N)
             ↓          ↓                ↓
          B(N + 1)   B(N + 2)   ...   B(2N − 1)
          ......................................
                                      A(N, N)
                                         ↓
                                    B(N(N+ 1)/2)
```

اكتب برنامجاً فرعياً SUBROUTINE لإنجاز ما سبق (هذه الطريقة يمكن بها تخزين مصفوفة متماثلة أو المثلث العلوى لمصفوفة) .

اجابات للمسائل التكميلية المختارة

٧ - ١٣ (ا) يجب ألا تكون هناك فصلة بعد FUNCTION أو عند نهاية الجملة لا يمكن أن تكون 2*X معامل .

(ب) البرنامج الفرعى FUNCTION ليس له اسم .

(جـ) لا يمكن أن تكون M(K) معامل ، ويجب ألا تكون هناك فصلة عند نهاية الجملة .

(د) لا يمكن أن تكون J(4) معامل .

٧ - ١٤ (ا) لا يمكن ان ... N — فى جملة DIMENSION.

(ب) لا يمكن أن تظهر K فى ... Y(K) حيث أن K ليست معامل .

٧ - ١٥ (ا) لم تخصص قيمة لـ DEP .

(ب) REAL ليس لها معنى بما أن LOOK لم تخصص لها قيمه .

٧ - ١٦ تطبع الأعداد الصحيحة 65 و 37 فى حقول بعرض 10 .

٧ - ١٧ (أ) حيث أن ADD لها قيمة صحيحة ، فيجب أن تعلن عنها فى جملة نوع :

```
     INTEGER FUNCTION ADD(M, N)
     ADD = 0
     DO 200 K = 1, N
          ADD = ADD + (M - 1) + K
200  CONTINUE
     RETURN
     END
```

(ب) الأعداد الصحيحة 14 و 42 و 34 تطبع على سطر واحد كل فى حقل بعرض 10 .

٧ - ١٨ (أ) لا نحتاج جملة نوع بما أن SUB لا تخصص له أى قيمة .

```
SUBROUTINE SUB(J, K, JSUM, JPROD, JDIFF)
JSUM = J + K
JPROD = J*K
JDIFF = J - K
RETURN
END
```

(ب) ستطبع الأعداد الصحيحة 14 ، 2 ، 5 ، 8 ، 33 على سطر .

٧ - ١٩ تطبع الأرقام الحقيقية 1.08 و 0.00 و 0.45 فى حقول بعرض 10.

٧ - ٢٠

```
F(R, S, T) = R*R + S*S - T*T
A = F(X, Y, 5.0)
B = F(2.0*X, 3.0*Y, Z)
C = SQRT(F(X, Y, 6.0))
```

الفصل الثامن

أساليب البرمجة والحسابات العددية

٨ ـــ ١ مقدمـــــة

توجد عادة عدة طرق لكتابة برنامج للتوصل إلى حل أو حلول لمسألة معينة . وتعتمد أفضل الطرق على عدد من العوامل ، ومنها أولا الحاجة إلى تقليل تأثير أخطاء التقريب حتى تكون النتائج دقيقة بقدر الإمكان . (انظر قسم ٢ — ١٠) — ثانياً أن يكون البرنامج على مستوى عال من الكفاءة بقدر الإمكان ، حيث تعتمد كفاءة البرنامج أساساً على عنصرين :

١ — الوقت المطلوب لتشغيل البرنامج .

٢ — عدد خلايا الذاكرة المستخدمة بواسطة البرنامج .

بمعنى آخر ، يجب أن نحاول كتابة البرنامج الذى يقلل وقت الحساب وعدد خلايا الذاكرة المستخدمة بواسطة البرنامج .

وعموما ، يمكننا كتابة برامج على مستوى عال من الكفاءة بتبنى عادات برمجة جيدة . فمثلا ، إذا ظهر تعبير عدة مرات فيجب أن يخصص له اسم بحيث تحسب قيمته مرة واحدة فقط ؛ فعند الرفع إلى اس مثل X**2.0 ، يجب أن يكتب فى الصورة X*X أو X**2 ويجب أن يكتب

3*X**2 + 4*X + 5 بدلا من 3.0*X**2 + 4.0*X + 5.0

رغم أن الحاسب يمكنه أن يقبل تعبيرات ذات نمط مختلط ، وهكذا ..

إلى جانب عادات البرمجة الجيدة ، يمكن إنقاص وقت الحساب بدرجة كبيرة باستخدام النظام الحسابى (الخوارزم) المناسب . على سبيل المثال إذا حسبت قيمة متعددة الحدود :

$$a_1X^N + a_2X^{N-1} + \cdots + a_NX + a_{N+1}$$

بالصورة المكتوبة لاحتاجت إلى عدد $N(N+1)/2$ من عمليات الضرب وإلى عدد N من عمليات الجمع . ولكنها ستحتاج إلى عدد N من عمليات الضرب وإلى عدد N من عمليات الجمع فقط باستخدام طريقة Horner (التى تناقش فى قسم ٨ — ٦) .

يمكن أيضاً إنقاص عدد أماكن الذاكرة بدرجة كبيرة باختيار النظام الحسابى (الخوارزم) المناسب . ولتوضيح ذلك افرض أننا نريد أن نعيد ترتيب عمود N من العناصر لمجموعة متراصة خطية A فى الترتيب العكسى ، أى ، بحيث تكون A(1) هى الأخيرة و A(2) قبل الأخيرة ، وهكذا . وأن تخزن العناصر مرة ثانية فى A إحدى الطرق لتنفيذ ذلك هى أن تخصص أولا عناصر A إلى مجموعة متراصة أخرى B (بها N عنصرا) بواسطة

$$B(N) \leftarrow A(1), \quad B(N-1) \leftarrow A(2), \quad \ldots, \quad B(1) \leftarrow A(N)$$

التي يمكن أن تنجز بواسطة حلقة DO التكرارية :

```
      DO 99 J = 1, N
            B(N + 1 – J) = A(J)
 99   CONTINUE
```

وتكون العناصر فى B فى الترتيب المطلوب . من ثم ، نعيد بعد ذلك العناصر إلى A مرة ثانية وذلك بتخصيص B(1) إلى A(1) ، B(2) إلى A(2) وهكذا . لاحظ أن هذه الطريقة تطلبت عددا إضافياً N من خلايا الذاكرة .

حقيقة يمكن أن تنجز مهمتنا باستخدام مكان ذاكرة واحد إضافى . أى أننا نستبدل A(1) ، A(N) ، A(2) و A(N — 1) وهكذا :

```
      NN = N/2
      DO 88 J = 1, NN
            TEMP = A(J)
            A(J) = A(N + 1 – J)
            A(N + 1 – J) = TEMP
 88   CONTINUE
```

وبذلك تكون الطريقة الثانية أكثر كفاءة من الطريقة الأولى . (مثال آخر لذلك يمكن أن نجده فى مناقشتنا فى قسم ٦ — ٨ حول تحريك جزء من مجموعة متراصة إلى أسفل مكان واحد) .

أخيرا ، يجب أن نشير إلى أهمية العامل البشرى فى عملية البرمجة . وبرغم أن كتابة 2*K فى صورة K + K قد توفر مقدار لا بأس به من وقت الحاسب إلا أنها قد تجعل البرنامج أكثر صعوبة فى الفهم وتتسبب فى ضياع الوقت البشرى . وبمعنى آخر ، من الأفضل أن يكون لدينا برنامج عام ومفهوم وسهل التنقل (مستقل عن الآلة) عن أن يكون لدينا برنامج معقد وغير مفهوم وملىء بخدع معتمدة على إمكانيات الآلة .

يعالج الجزء الأول من هذا الفصل تقنيات البرمجة الفنية المتعددة لمسائل معينة تظهر مراراً ، مثل الفرز والبحث والإدماج . ويخصص الجزء الثانى من هذا الفصل لحسابات الرقمية مثل حل المعادلات الخطية وإيجاد جذور متعددة الحدود وضرب المصفوفات .

٨ — ٢ الفــــــرز

نعنى بالفرز ، ترتيب العناصر فى نظام ما . هذا الإجراء معتاد فى الحياة اليومية فالأسماء فى دليل التليفون مفروزة أبجدياً . ويمكن أن تفرز سجلات الطلبة حسب أرقام ID وهكذا ، ورغم أن الفرز يمكن أن يبدو كما لو كان مهمة تافهة . ولكن يمكن أن يكون الفرز بكفاءة صعباً جداً من الناحية العملية والنظرية .

لما كان الفرز والبحث يستخدمان فى الاحتفاظ بملفات عمليات تشغيل البيانات نبدأ أولا بتقديم بعض الاصطلاحات . يفرز ملف السجلات عند تشغيل البيانات تبعاً لمفتاح معين . على سبيل المثال ، تحتفظ الجامعة بملف لطلبتها وتسمى البيانات الخاصة بكل طالب ، فى الملف سجل . وقد يحتوى السجل على عدة حقول . على سبيل المثال ، الاسم ورقم الضمان الاجتماعى ورقم الطالب والنوع والفرقة والمتوسط التراكمى للدرجات وهكذا . عند فرز الملف تبعاً للترتيب الأبجدى للأسماء يكون حقل الاسم هو المفتاح أما إذا كان الفرز تبعاً لرقم الضمان الاجتماعى فيكون رقم الضمان الاجتماعى هو المفتاح .

وبصدد هذا ، سندرس مشكلة فرز مجموعة متراصة من الأرقام A(1) ، A(2)....A(N) وليكن فرزاً تصاعدياً ، حتى يكون

$$A(1) < A(2) < A(3) < \cdots < A(N)$$

سوف نفترض أن لدينا توصلا فورياً للعناصر وأن هذه العناصر مخزنة فى الذاكرة الأساسية ويسمى هذا فرزاً داخلياً .

أحياناً تكون الملفات كبيرة جداً بحيث تحتفظ بها على وحدات التخزين الخلفية بتوصل محدود . ويسمى فرز من هذا النوع من الملفات الفرز الخارجى وهو صعب ويقع خارج حدود هذا الكتاب .

(أ) الفرز الفقاعى

إحدى طرق فرز A تكون بواسطة النظام الحسابى (حوارزم) المسمى فرز فقاعى والذى تمت مناقشته فى المسألة ٦ — ١ . أى ، نقارن أولا A(1) و A(2) وترتبهما فى النظام المطلوب حيث يكون A(1) < A(2) ثم نقارن A(2) و A(3) وترتبهما بحيث يكون A(2) < A(3) ثم نقارن A(3) و A(4) وترتبها بحيث يكون A(3) < A(4) نكمل العمل حتى نقارن A(N — 1) و A(N) وترتبهما بحيث تكون A(N — 1) < A(N) بعد هذا المسح (أو المرور) خلال العناصر باكلها يقفز العنصر الأكبر إلى أعلى ، أى ، إلى المكان N .

نكرر الإجراء السابق للعناصر A(1) ، A(2) ، .....A(N — 1) بعد المسح (المرور) خلال هذه العناصر يقفز العنصر الأكبر إلى أعلى ، أى إلى المكان N — 1 . وهكذا نكمل هذا الإجراء ، وبعد عدد N — 1 من المرات سيتم فرز المجموعات المتراصة A تصاعدياً . تظهر خريطة سير العمليات للنظام الحسابى (الخوارزم) وترجمته للفورتران فى مسألة ٦ — ١٢ .

يمكن تحسين كفاءة هذا النظام الحسابى (الخوارزم) كما يلى . نلاحظ أولا أنه من خلال كل عملية مسح (أو مرور) (حلقة DO الخارجية) ، يحدث التبديل فقط عندما تكون هناك عناصر ليست مرتبة فى النظام المطلوب . وتبعاً لذلك يمكن إضافة عداد لعد عدد التبديلات فى كل عملية مسح (أو مرور) ، وإن لم يحدث تبديل فسوف تكون المجموعة المتراصة قد تم فرزها وتكون المهمة قد انتهت .

(ب) فرز انتقائى

يسمى النظام الحسابى (الخوارزم) الثانى للفرز ، الفرز التبديلى أو الفرز الانتقائى ويعمل كالتالى : أولا نجد القيمة الصغرى فى المجموعة المتراصة ونضعها فى المكان الأول . ثم نجد القيمة الصغرى التى تليها ونضعها فى المكان الثانى ، وهكذا . وفيما يلى نعرض بدقة أكثر هذا النظام الحسابى (الخوارزم) .

(١) أوجد المكان L لأصغر عنصر بين العناصر A(1) ، A(2) ، .... A(N) ثم استبدل A(L) مع A(1)

(٢) أوجد المكان L لأصغر عنصر بين A(2) ، A(3).......A(N) ثم استبدل A(L) مع A(2)

... ...

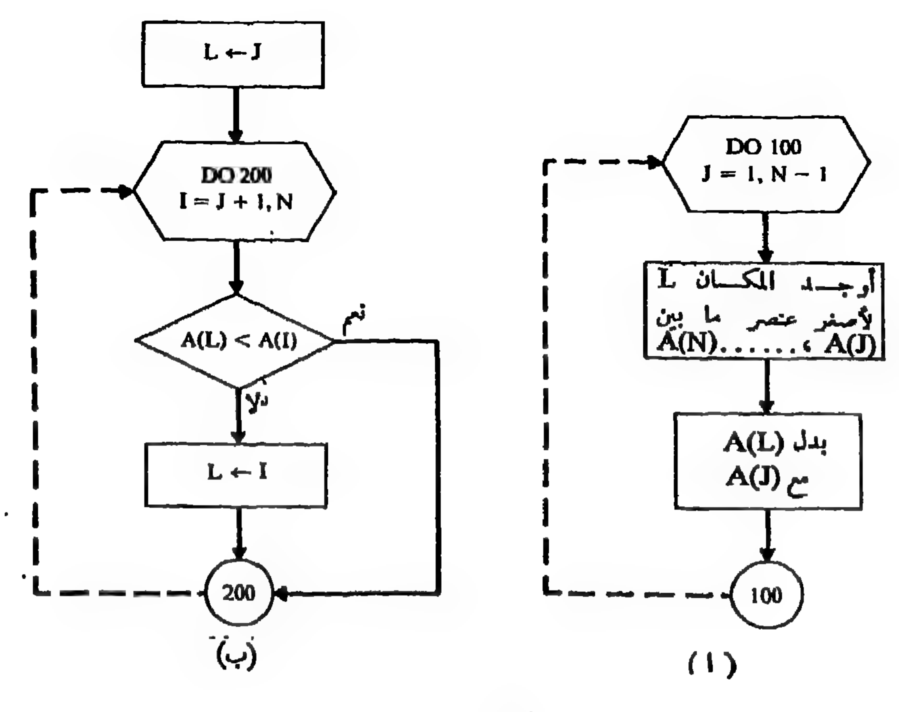
(N — 1) أوجد المكان L لأصغر عنصر بين A(N — 1) و A(N) ثم استبدل A(L) مع A(N — 1)

بذا تصبح المجموعة المتراصة A مفروزة فى الترتيب التصاعدى .

لاحظ أن الخطوات متماثلة مشابه لبعضها ، من ثم ، فمن الواضح أننا نستطيع أن نستخدم حلقة DO محكومة بدليل J يتغير من 1 إلى N — 1 . بداخل حلقة DO هذه نريد أن نجد المكان L لأصغر عنصر بين A(J) ، ....A(N) وعندئذ نستبدل A(L)

مع A(J) . تظهر خريطة سير العمليات المصغرة لحلقة DO في شكل ٨ – ١ (ب) . ينتج عن إدماج خريطتي سير العمليات هذه برنامج واحد وهو برنامج الفرز المطلوب الآتي :

```
C
C         SORTING PROGRAM FOR A(1), . . . , A(N)
C
          NN = N - 1
          DO 100 J = 1, NN
C
C                 FIND LOCATION L OF SMALLEST
C
                  L = J
                  JJ = J + 1
                  DO 200 I = JJ, N
                          IF(A(L).LT.A(I)) GO TO 200
                          L = I
  200             CONTINUE
C
C                 INTERCHANGE A(L) WITH A(J)
C
                  T = A(L)
                  A(L) = A(J)
                  A(J) = T
  100     CONTINUE
```



شكل ٨ – ١

ملحوظة : يمكن حساب عدد المقارنات فى الفرز الفقاعى كالتالى . خلال المرور الأول نجد أن هناك عدد (N — 1) من المقارنات ، وخلال المرور الثانى نجد أن هناك عدد (N — 2) من المقارنات ، وهكذا . وبذلك يكون ، العدد الكلى للمقارنات هو :

$$(N-1)+(N-2)+\cdots+2+1=\frac{N(N-1)}{2}$$

أما بالنسبة للفرز الانتقائى ، فنحتاج إلى عدد (N — 1) من المقارنات وذلك لإيجاد العنصر الأصغر . ولإيجاد العنصر الأصغر الذى يليه نحتاج إلى عدد (N — 2) من المقارنات ، وهكذا . ، لذلك يكون المجموع المطلوب مرة أخرى هو عدد N(N —1)/2 من المقارنات . نلاحظ أن متوسط عدد التباديل سوف يكون N(N — 1)/4 وهو نصف عدد المقارنات ، يمكن أن ننقص عدد المقارنات بواسطة التعديل المناسب (انظر مسألة ٨ — ١) . ولكن قد يبقى متوسط هذه التباديل كما هو .

٨ — ٣ الادمــــاج

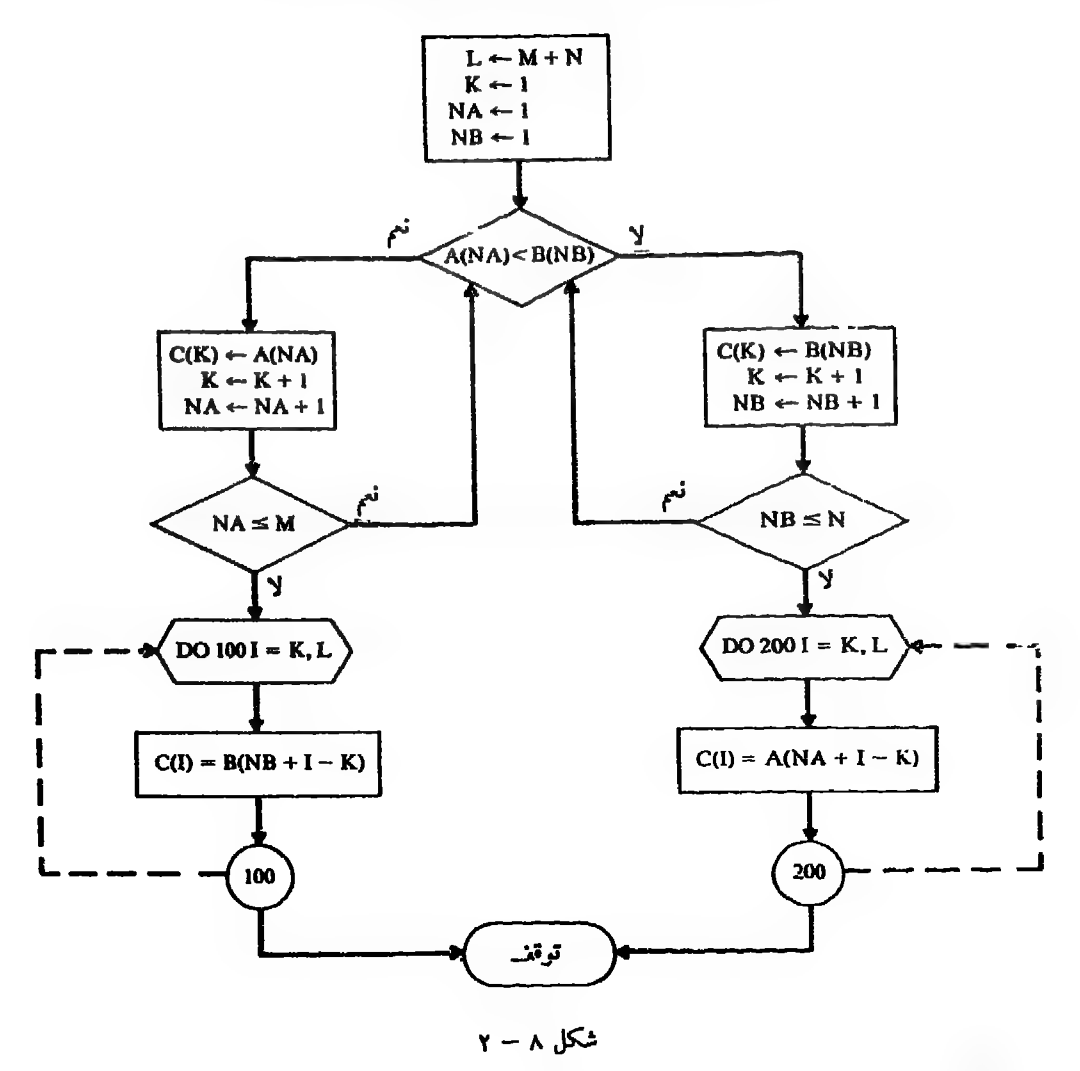
افرض أن A و B مجموعات متراصة مفروزة (ولتكن فى الترتيب التصاعدى) بهما عدد M و N من العناصر على الترتيب . ونود أن ندمج A و B وذلك بتكوين مجموعة متراصة مفروزة C إحدى الطرق لعمل ذلك هى نقل A ، B إلى C عشوائيا ، وليكن A فى البداية و B فى النهاية ، ثم نفرز C كما فى قسم ٨ — ٢ . لاحظ أن هذه الطريقة لا تستعمل حقيقة أن A ، B مفروزان فعلا . وسوف نعطيك برنامج أكثر كفاءة فيما بعد .

دعنا نحفز أولا النظام الحسابى (الخوارزم) الخاص بنا . افرض أن لدينا صفين من الطلبة مفروزان تصاعدياً حسب الطول ، وافرض أننا نريد دمجهما فى صف واحد مفروز . إحدى الطرق لتنفيذ ذلك يتحقق بملء الصف الجديد تتابعياً بالطالب الأقصر طولا بين الطالبين اللذين يقفان فى مقدمتى الصفين الأصليين إلى أن ينتهى أحد الصفين (أى عندما لا يوجد به طلبه آخرون) . ثم ينضم الطلبة الباقون فى نهاية الصف المدموج .

من أجل ترجمة هذا النظام الحسابى (الخوارزم) إلى فورتران ، يجب أن نتتبع دائماً أماكن العناصر الأصغر من A ، B التى لم توضع بعد فى C . سنجعل NA ، NB تشير إلى هذه الأماكن ، على الترتيب ، وبذلك عند كل خطوة نسأل السؤال التالى :

$$A(NA) < B(NB)?$$

إذا كانت الإجابة نعم ، نضع A(NA) فى C ونزيد NA بمقدار 1 وإذا كانت الإجابة لا ، نضع B(NB) فى C ونزيد NB بمقدار 1 ، إذا وصلت NA أو NB إلى قيمها العظمى (أى M أو N على الترتيب) عندئذ . نضع العناصر الباقية من المجموعة المتراصة الأخرى فى C خريطة سير العمليات تظهر فى شكل ٨ — ٢ .



شكل ٨ — ٢

٨ — ٤ البحث

افرض أن A مجموعة متراصة خطية مخزنة فى الذاكرة ، ونريد أن نجد المكان L للعنصر D فى هذه المجموعة المتراصة .

(ا) البحث المتتالى (البحث الخطى)

يمكننا أن نبحث ببساطة عن D مع عدم إعطاء أى معلومات إضافية عن A وذلك عن طريق مقارنتها بالعناصر الموجودة فى A عنصراً عنصراً . تسمى هذه الطريقة بطريقة البحث المتتالى . أو البحث الخطى وقد تم توصيفها فى المسألة ٦ — ١١ . من الممكن أن نرى بسهولة ، أن هذه الطريقة تتطلب عدد N من المقارنات لإيجاد D فى أسوأ الحالات وعدد N/2 من المقارنات فى المتوسط .

(ب) البحث الثنائى

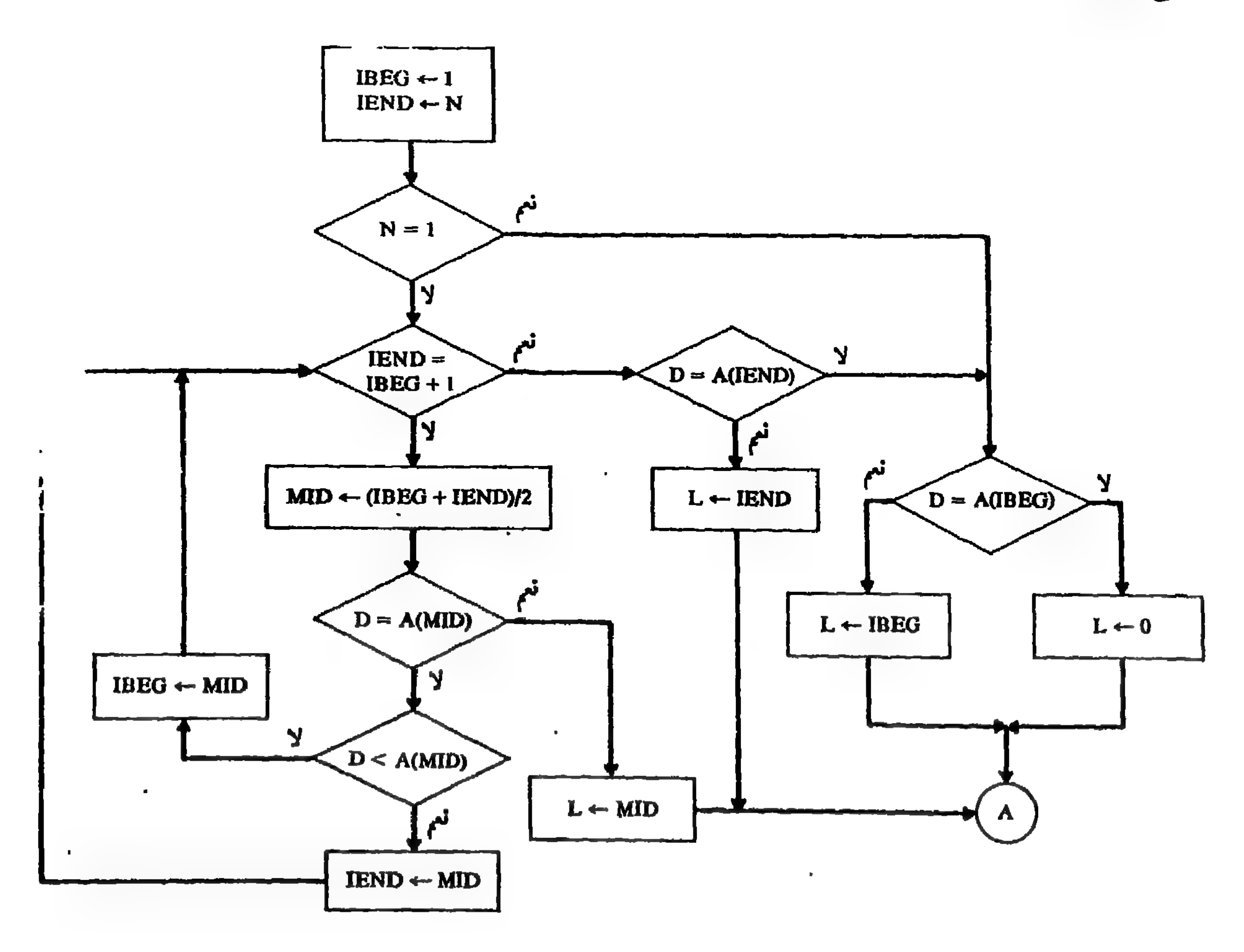
افرض الآن أن A مجموعة متراصة مفروزة . حيث أن طريقة البحث الخطى لا تتأثر بحقيقة أن A مفروزة ، فإنه يمكن استخدام نظام حسابى (خوارزم) أكثر كفاءة ، يسمى البحث الثنائى ، وذلك للبحث عن D .

دعنا نحفز أولا النظام الحسابى الخاص بنا . افرض أننا نبحث عن اسم فى دليل التليفون حيث الأسماء مرتبة أبجدياً . فنحن لا نبحث عن الاسم على التوالى من أول الصفحة إلى أن نجد المطلوب إنما نفتح الدليل من الوسط لنقرر أى نصف من الدليل يحتوى على الاسم ثم نفتح النصف من الوسط لنقرر أى ربع من الدليل يحتوى على الاسم ، وهكذا إلى أن نجد الاسم المطلوب .

سنصف الآن النظام الحسابي (الخوارزم) للبحث الثنائي بصورة رسمية . أولا اقسم المجموعة المتراصة A إلى جزئين :
A(1) ، A(2) ، ....A(MID) و A(MID) ، A(MID + 1) ......A(N)
حيث $MID = (1 + N)/2$ أي :
(١) $IBEG = 1$
(٢) $IEND = N$
(٣) $MID = (IBEG + IEND)/2$
(٤) إذا كانت $D = A(MID)$ ، من ثم $L = MID$ ، وبذلك نكون قد وصلنا .
(٥) إذا كانت $D < A(MID)$ من ثم تكون D فى النصف الأول من المجموعة المتراصة وإلا تكون D فى النصف الثانى من المجموعة المتراصة .

لاحظ أنه إذا كانت D فى النصف الأول من المجموعة المتراصة ، من ثم تكون A(MID) هى نهاية هذا النصف . لكن إذا كانت D فى النصف الثانى من المجموعة المتراصة فسوف تكون A(MID) هى بداية هذا النصف . وتنشىء هذه الفكرة حلقة تكرارية وذلك بإعادة كتابة خطوة (٥) كما يلى :

(٥) إذا كانت $D < A(MID)$ إذاً أعد وضع IEND إلى MID واذهب إلى (٣) وإذا لم يتحقق ذلك فضع IBEG إلى MID ، اذهب إلى (٣)



شكل ٨ — ٣

يعمل الإجراء السابق جيداً طالما يحتوى الجزء A(IBEG) ، . . . ، A(IEND) من المجموعة المتراصة فى أى لحظة على أكثر من عنصرين . إذا كان الجزء به عنصرين ، أى ، إذا كانت IEND = IBEG + 1 فسوف تكون قيمة MID هى :

(IBEG + IBEG + 1)/2

سوف تكون النتيجة IBEG مرة ثانية وذلك بسبب القسمة الصحيحة. من ثم ، سيكون لدينا حلقة تكرارية لا نهائية إذا كانت D>A(IBEG). وافرض ثانيا نفحص IBEG ، IEND لنرى إذا كانت أرقام صحيحة متتالية .

نقطة أخيرة . وهى إذا كانت N = 1 أى ، إذا ابتدأت المجموعة المتراصة أصلا بعنصر واحد فسوف تحدث حلقة تكرارية لا نهائية أيضاً إذا كانت D لا تنتمى إلى المجموعة المتراصة وبذلك يجب أن نفحص أولا لنرى ما إذا كانت N = 1 . تظهر خريطة سير العمليات فى شكل ٨ — ٢ وترجمتها إلى الفورتران متروكة كتدريب الطالب (مسألة ٨ — ٣) .

ملاحظة : نختم هذا القسم بمقارنة طريقة البحث الثنائى مع طريقة البحث المتتالى — باستخدام طريقة البحث الثنائى ، تحذف البدائل تقريباً فى كل مرة تنفذ فيها الحلقة التكرارية . (انظر شكل ٨ — ٣) . أى إذا أعطينا عدد N من الاختيارات الأصلية فسوف تبقى هناك عدد N/2 من الاختيارات بنهاية اللفة الأولى . وتبقى عدد (N/2)/2 بنهاية اللفة الثانية . وتبقى عدد N/8 = (2*2*2)/N من الاختيارات بنهاية اللفة الثالثة . وهكذا . فالعدد الكلى للمقارنات فى البحث الثنائى فى أسوأ الحالات هو $Log_2 N$ وهو تحسن عظيم إذا قورن بالبحث المتتالى . فى الحقيقة $Log_2 N$ أصغر كثيراً من N/2 وهو متوسط عدد المقارنات فى البحث المتتالى . على سبيل المثال ، إذا كانت N = 10,000 فإن N/2 = 5000 بينما $\log_2 10{,}000 \approx 14$

٨ — ٥ التحـــــديث

افرض أن A(1) ، A(2) ، . . . .A(N) مخزنة فى الذاكرة ، وأن D ليست فى المجموعة المتراصة A ونريد أن نضيف العنصر D إلى هذه المجموعة المتراصة .

إذا كانت عناصر A مخزنة عشوائياً ، إذن فيمكننا ببساطة اضافة D إلى نهاية المجموعة المتراصة بواسطة جملة التخصيص D → A(N + 1) أى بتخصيص D إلى A(N + 1) . أما إذا كانت A مجموعة متراصة مفروزة فيجب أولا أن نجد المكان L الذى يجب أن تضاف فيه D وذلك قبل إضافة D إلى المجموعة المتراصة . وهذا يكافئ مرة أخرى نفس مهمة البحث ومع تعديلات طفيفة يمكن استخدام البحث الثنائى لإيجاد المكان L الذى تمت مناقشته سابقاً (انظر مسألة ٨ — ٦) .

هل يمكننا ببساطة تخصيص D إلى A(L) بمجرد تحديد المكان L ؟ ، واضح أن الإجابة هى لا ، حيث أن هذا سيؤدى إلى إزالة A(L) . لذلك ، يجب أولا أن نخلى المكان بتحريك جزء المجموعة المتراصة A(L + 1) ، . . . . .A(N) أسفل القائمة بموضع واحد :

A(L) → A(L + 1) → . . . → A(N − 1) → A(N) → A(N + 1)

الخطوات موصوفة بالكامل فى المسألة ٦ — ١٠ .

٨ — ٦ طريقة هـــورنر

ادرس الدالة كثيرة الحدود التالية :

$$f(x) = 3x^4 - 5x^3 + 6x^2 + 8x - 9$$

لاحظ أن هناك عدد 4 + 3 + 2 + 1 = 10 من عمليات الضرب و 4 عمليات جمع . إفرض الآن ، أننا أعدنا كتابة كثيرة الحدود بالتحليل المتتالى لـ X كما يلى :

$$
\begin{aligned}
f(x) &= (3x^3 - 5x^2 + 6x + 8)x - 9 \\
&= ((3x^2 - 5x + 6)x + 8)x - 9 \\
&= (((3x - 5)x + 6)x + 8)x - 9
\end{aligned}
$$

بذلك تتطلب كثيرة الحدود 4 عمليات ضرب و 4 عمليات جمع فقط .تستخدم هذه الطريقة لإيجاد قيمة كثيرة الحدود وتسمى طريقة هورنر ، ويتم وصفها بطريقة عامة كما يلى :

ادرس الآن المعادلة العامة لكثيرة الحدود ذات الدرجة N :

$$Y = F(X) = A_1X^N + A_2X^{N-1} + \cdots + A_NX + A_{N+1}$$

سوف يتطلب إيجاد قيمة كثيرة الحدود عدد مساو لـ

$$N + (N - 1) + (N - 2) + \cdots + 2 + 1 = N(N - 1)/2$$

من عمليات الضرب وعدد N من عمليات الجمع . ومع ذلك إذا أوجدنا قيمة كثيرة الحدود باستخدام طريقة هورنر :

$$Y = F(X) = ((\ldots((A_1X + A_2)X + A_3)X + \cdots)X + A_N)X + A_{N+1}$$

فسوف نحتاج فقط إلى عدد N من عمليات الضرب وعدد N من عمليات الجمع . وبذلك تكون هذه الطريقة أكثر كفاءة .

[طر]يقة هورنر ميزة أخرى هامة . لاحظ أنه عند كل خطوة فى إيجاد قيمة Y ، نضرب فى X ونجمع المعامل التالى . وبه يمكن أن نستخدم حلقة DO لإيجاد قيمة Y كما يلى :

```
      Y = A(1)
      NN = N + 1
      DO 100 J = 2, NN
           Y = Y*X + A(J)
100   CONTINUE
```

فى الحقيقة ، يمكن أن نكتب برنامج فرعى لإيجاد قيمة كثيرة الحدود من أى رتبة (انظر مسألة ٨ – ١٣) .

ملاحظة : افرض أننا جعلنا B(1) ، B(2) ، ... ، B(N + 1) تشير إلى النتائج الجزئية فى طريقة هورنر بمعنى :

$$B(1) = A(1), \quad B(2) = B(1)*X + A(2), \quad \ldots, \quad B(N + 1) = B(N)*X + A(N + 1)$$

وعلى ذلك يمكن أن نبين أن قيمة التفاضل F'(X) هى :

$$F'(X) = B_1X^{N-1} + B_2X^{N-2} + \cdots + B_{N-1}X + B_N$$

والتى يمكن أيضاً أن نوجد قيمتها بواسطة طريقة هورنر . فى الحقيقة جزء البرنامج التالى يعطى قيم F(X) و F′(X) :

```
      B(1) = A(1)
      NN = N + 1
      DO 100 J = 2, NN
            B(J) = B(J - 1)*X + A(J)
100   CONTINUE
      C(1) = B(1)
      DO 200 K = 2, N
            C(K) = C(K - 1)*X + B(K)
200   CONTINUE
```

فى هذا الجزء من البرنامج F(X) = B(N + 1) و F′(X) = C(N)

٨ ــ ٧ حل معــادلات معينــة

يناقش هذا القسم إيجاد حلول حقيقية للمعادلات التى تنشأ من الدوال الكثيرة الحدود والمثلثية واللوغاريتمية ، والأسية ، وعموماً ، هناك وجهتان للمشكلة :

١ ــ إيجاد المنطقة المجاورة للجذر .

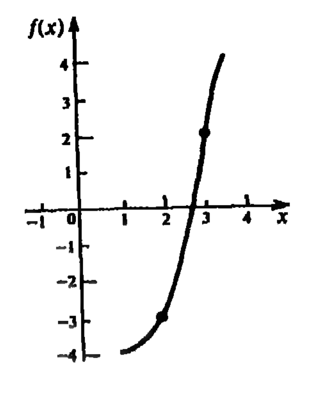
٢ ــ تقريب الجذر .

لأسباب تعليمية خاصة نناقش أولا طرق تقريب الجذر إذا كانت المنطقة المجاورة معروفة .

ادرس المعادلة كثيرة الحدود التالية :

$$f(x) = x^3 - 4x^2 + 6x - 7 = 0$$

لاحظ أن $f(2) = -3$ سالبة $f(3) = 2$ موجبة . وحيث أن f هى دالة كثيرة الحدود ورسمها البيانى لا يشتمل على أى نقاط عدم استمرارية فى المنحنى الخاص بها ، وعلى ذلك يجب أن يقطع الرسم البيانى للدالة f محور x ما بين 2 ، 3 كما هو موضح فى شكل ٨ ــ ٤ . والنقطة التى يقطع عندها الرسم البيانى محور x هى حل حقيقى للمعادلة . ونعطى هنا النظام الحسابى (الخوارزم) المعروف باسم (تنصيف المدى) والذى يحدد قيمة تقريبية لمثل هذا الجذر ، أى عدد يقل عن الجذر الحقيقى بأقل من 0.001



شكل ٨ ــ ٤ رسم $f(x)$ بيانياً .

(١) (تنصيف المدى)

أولا ضع XN = 2.0 و XP = 3.0 أى XN هى النقطة التى عندها f سالبة ، و XP هى النقطة التى عندها f موجبة . تحدد النقطة الوسطى XM بين XN و XP بالمعادلة التالية :

$$XM = (XN + XP)/2$$

تحدد النقطة الوسطى XM وتوجد قيمة $f(XM)$:

$XM = (XN + XP)/2 = 2.5$ و، $f(XM) = f(2.5) = -1.375$

حيث أن $f(XM)$ سالبة ، ضع $XN = 2.5$. ثم أوجد XM و $F(XM)$ مرة أخرى :

$XM = 2.75$ و $f(XM) = f(2.75) = 0.046875$

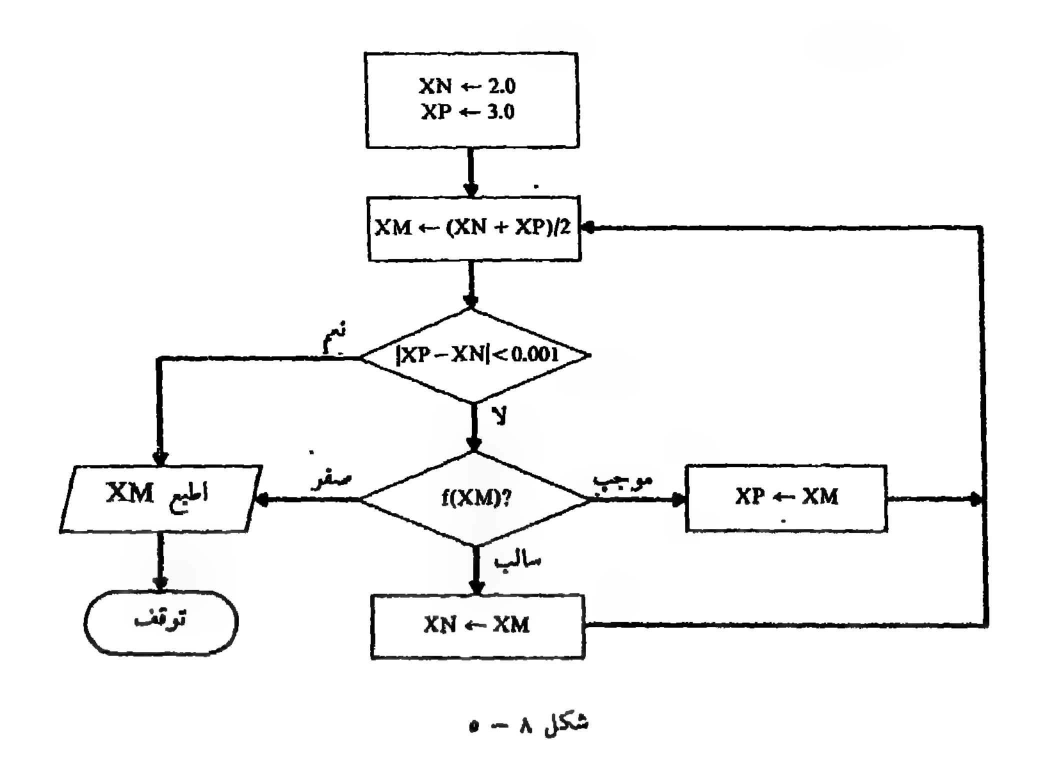
حيث أن $f(XM)$ موجبة نضع $XP = 2.75$ لاحظ أن الجذر يقع الآن بين 2.5 و 2.75

نكرر الإجراء السابق . إذا حصلنا على $f(XM) = 0$ عند أى خطوة فإن XM هو جذر . فيما عدا ذلك ، فإن XP ، XN يقتربان من بعضهما أكثر فأكثر . فإذا كانت

$$|XP - XN| < 0.001$$

فإن قيمة XM الجديدة هى قيمة مقربة للجذر الحقيقى فى حدود 0.001 .

تظهر خريطة سير العمليات للنظام الحسابى (الخوارزم) فى شكل ٨ — ٥ .



شكل ٨ — ٥

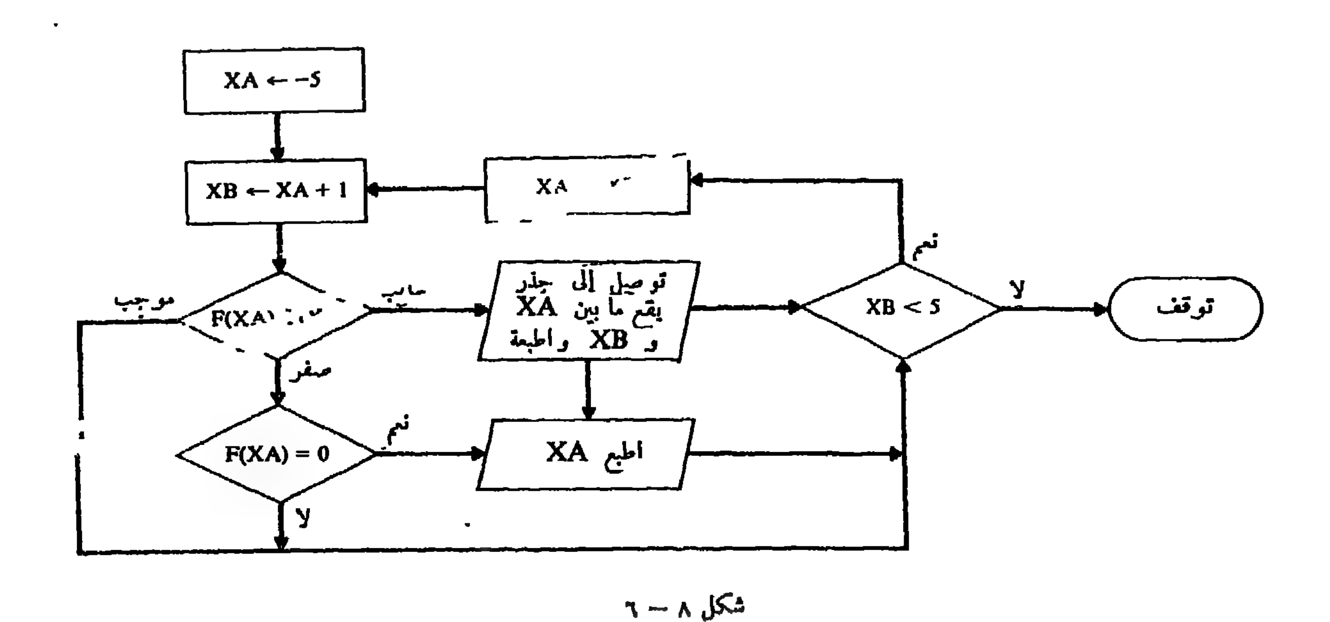
(ب) مواقع الجذور

يعمل النظام الحسابى (الخوارزم) السابق بصورة جيدة إذا أعطينا الدالة $f(x)$ والنقط A ، B التى عندها $f(A)$ و $f(B)$ لهما إشارات مختلفة . ولكن المهمة الأكثر صعوبة هى إيجاد هاتين النقطتين A و B . سنناقش هذا فيما يلى :

افرض أننا اعطينا دالة $f(x)$ نبحث عن جذر لها وليكن ما بين $x = -5$ و $x = 5$ يمكن أن نحدد قيمة $f(x)$ عند $x = -5, -4, \ldots 5$ لو تغيرت إشارة الدالة f عند أى زوج من القيم K و $K + 1$ فنعرف أن $f(x)$ لها جذر بين K و $K+1$ ويمكن أن نكمل كما فى (أ) وبالتحديد نضع أولا $XA = -5$ و $XB = XA + 1 = -4$ ونختبر حاصل الضرب .

F(XA)*F(XB)

إذا كان حاصل الضرب سالباً فإن إشارة F(XA) و F(XB) تكون مختلفة وبذا نحدد ونطبع جذر يقع ما بين XA و XB ومن ناحية أخرى ، إذا كان حاصل الضرب موجباً فإن كلا من F(XA) و F(XB) لهما نفس الإشارة ، ولا نعرف إذا ما كان هناك جذرا يقع ما بين XA و XB أم لا . فى كلتا الحالتين ، نعيد تحديد قيمة XA لتكون $XA = -4$ ثم نعيد تحديد قيمة XB عند $XA + 1 = -3$ ونكرر نفس الإجراء ينتهى التكرار عند $XB = 5$. يعطى الشكل ٨ – ٦ خريطة سير العمليات لمثل هذا النظام الحسابى (الخوارزم) تأخذ أيضاً خريطة سير العمليات فى الاعتبار احتمال أن يكون حاصل الضرب F(XA) × F(XB) صفرا ، وفى تلك الحالة قد تكون XA جذرا .



شكل ٨ – ٦

من المهم أن نتذكر أن من الممكن أن تظل $f(x)$ لها جذور حقيقية رغم أن إشارتها لا تتغير عند النقط ، على سبيل المثال ، فالدالة :

$$f(x) = 8x^2 - 26x + 21$$

لها جذران يقعان بين 1 و 2 ومع ذلك فهى موجبة عند كل النقط الصحيحة . وعلى ذلك قد يتطلب الأمر إيجاد قيمة $f(x)$ باستخدام قيم أصغر أو فى مدى أوسع قبل أن نيأس من إيجاد الحل . وستكون أى مناقشة أعمق لهذه المشكلة واقعة خارج نطاق هذا الكتاب .

٨ — ٨ التكامل العددى

افرض دالة f موجبة فى المدى $D = \{ a \le x \le b \}$ ، والمطلوب إيجاد المساحة تحت المنحنى $y = f(x)$ ما بين $x = a$ و $x = b$ هذه المساحة مظللة فى شكل ٨ — ٧ وتساوى :

$$\int_a^b f(x)\,dx$$

أى التكامل للدالة f من a إلى b

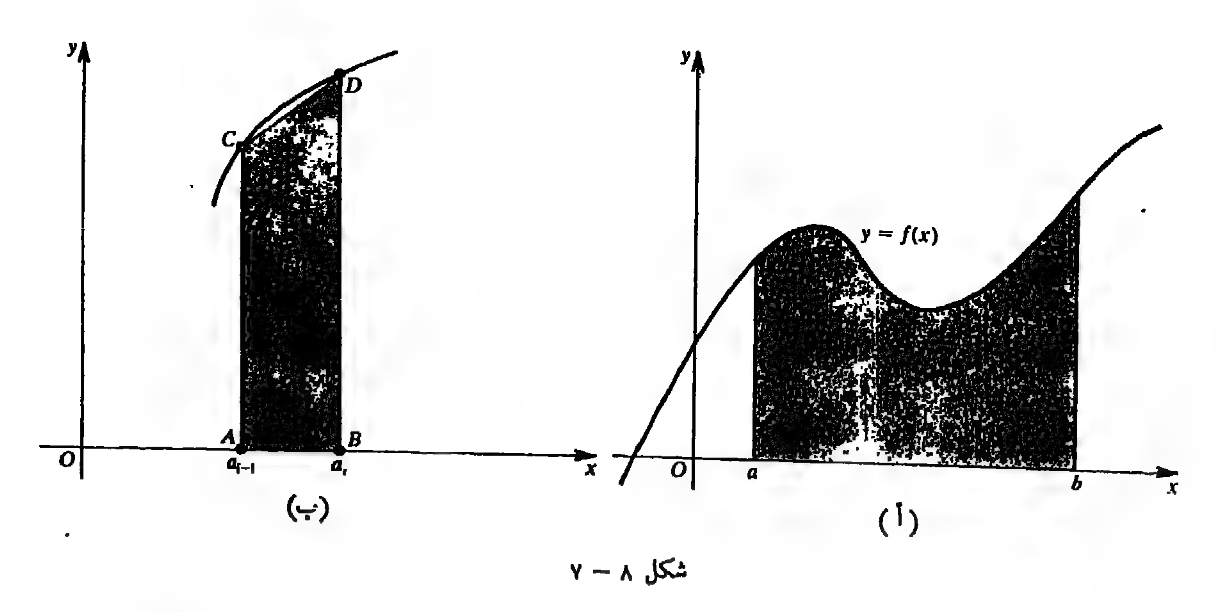
أولا نقسم D إلى عدد n من الأجزاء المتساوية كل بطول $h = (a - b)/n$ وذلك باختيار النقط :

$$a_0 = a, \quad a_1 = a + h, \quad a_2 = a + 2h, \quad \ldots, \quad a_{n-1} = a + (n-1)h, \quad a_n = a + nh = b$$

ولنفرض أن المعادلة الآتية تعطى تقريب المساحة :

$$\text{area} \approx \tfrac{1}{2}h[f(a_0) + 2f(a_1) + 2f(a_2) + \cdots + 2f(a_{n-1}) + f(a_n)]$$

تسمى هذه العلاقة « قاعدة شبه المنحرف » فى ضوء مشتقاتها التى نصفها فيما يلى :



شكل ٨ — ٧

لاحظ أنه عندما تكون n كبيرة و h صغيرة ، فالمنحنى $y = f(x)$ ما بين $x = a_{i-1}$ و $x = a_i$ هو تقريباً خط مستقيم كما هو مصور فى شكل ٨ — ٧ (ب) وبالتالى فالمساحة تحت المنحنى تكون أقرب إلى مساحة شبه المنحرف T_i برؤوس A و B و D و C لاحظ أن :

$$AC = f(a_{i-1}) \qquad \text{and} \qquad BD = f(a_i)$$

وأيضا h هى المسافة بين الجانبين AC و BD . من ثم :

$$\text{area}(T_i) = \tfrac{1}{2}h[f(a_{i-1}) + f(a_i)]$$

وتبعاً لذلك ، فالمساحة تحت المنحنى مساوية لمجموع مساحات أشباه المنحرفات تقريباً :

$$\text{area} \approx \text{area}(T_1) + \text{area}(T_2) + \cdots + \text{area}(T_n)$$

وبالتالى فهى تحدد العلاقة المطلوبة .

تكون الدالة $f(x) = 3x^3 - 4x^2 + 6x + 5$ موجبة كلما كانت x موجبة . يقرأ البرنامج التالى A و B و N حيث $0 < A < B$ وكذلك يحسب المساحة التقريبية بقاعدة شبه المنحرف ، ويطبع A و B والمساحة التقريبية .

```
C        AREA BY TRAPEZOID RULE
         F(X) = ((3.0*X - 4.0)*X + 6.0)*X + 5.0
         READ(5, 10) A, B, N
    10   FORMAT(2F8.2, I5)
         H = (B - A)/FLOAT(N)
         SUM = F(A) + F(B)
         NN = N - 1
         DO 200 K = 1, NN
                  SUM = SUM + 2*F(A + FLOAT(K)*H)
   200   CONTINUE
         AREA = H*SUM/2.0
         WRITE(6, 20) A, B, AREA
    20   FORMAT(1X, 2F10.2, F15.3)
         STOP
         END
```

٨ ــ ٩ المتجهات والمصفوفات

يبحث هذا القسم فى العمليات الجبرية التى تجرى على المتجهات والمصفوفات . وفى التصوير الرياضى فالمتجه هو مجموعة مترّاصة خطية (أى ذات بعد واحد) ، والمصفوفة هى مجموعة متراصة ذات بعدين . (فى مفهوم المتجهات والمصفوفات ، يستخدم مصطلح اللا متجه للأرقام المفردة) .

(أ) المتجهات

تشير الرياضيات عادة إلى المتجه A (مجموعة متراصة خطية) بواسطة :

$$A = (A_1, A_2, \ldots, A_n)$$

على سبيل المثال ، تشير :

$$A = (2, -3, 5, 7)$$

إلى أن A متجه به 4 عناصر هى 2 و 3ــ و 5 و 7

افرض A و B متجهين كل به عدد n من العناصر ، و X قيمة غير متجهة ، بمجموع A و B هو المتجه الذى نحصل عليه بجمع العناصر المناظرة فى B, A .

$$A + B = (A_1 + B_1, A_2 + B_2, \ldots, A_n + B_n)$$

وحاصل ضرب المتجه A فى القيمة غير المتجهة X هو المتجه الذى نحصل عليه بضرب كل عنصر من A فى X .

$$X \cdot A = (XA_1, XA_2, \ldots, XA_n)$$

حاصل الضرب الداخلى (نقطة غير متجهة) لـ A و B يكتب بالصورة A.B وهى القيمة غير المتجهة :

$$A \cdot B = A_1B_1 + A_2B_2 + \cdots + A_nB_n = \sum_{k=1}^{n} A_kB_k$$

طول A (أو : norm) ، يكتب $||A||$ هو القيمة غير المتجهة :

$$\|A\| = \sqrt{A \cdot A} = \sqrt{A_1^2 + A_2^2 + \cdots + A_n^2}$$

على سبيل المثال ، افرض $A(2, -3, 5, 7)$ و $B(4, 2, -1, 3)$ ، وعلى ذلك :

$$A + B = (2 + 4, -3 + 2, 5 - 1, 7 + 3) = (6, -1, 4, 10)$$
$$3A = (3 \cdot 2, 3 \cdot (-3), 3 \cdot 5, 3 \cdot 7) = (6, -9, 15, 21)$$
$$A \cdot B = 2 \cdot 4 + (-3) \cdot 2 + 5 \cdot (-1) + 7 \cdot 3 = 8 - 6 - 5 + 21 = 18$$
$$\|A\|^2 = 2^2 + (-3)^2 + 5^2 + 7^2 = 4 + 9 + 25 + 49 = 87, \text{ and so } \|A\| = \sqrt{87}$$
$$\|B\| = \sqrt{16 + 4 + 1 + 9} = \sqrt{30}$$

افرض A و B و C و D متجهات خطية كل بها عدد N من العناصر و X قيمة غير متجهة . فيما يلى جزء من برنامج يخزن مجموع A و B فى C ويخزن حاصل الضرب غير المتجه X و A فى D ويخزن الكميات غير المتجهة A.B و $||A||$ و $||B||$ فى DOT و ANORM و BNORM على الترتيب :

```
      DOT = 0.0
      SUMA = 0.0
      SUMB = 0.0
      DO 100 K = 1, N
            C(K) = A(K) + B(K)
            D(K) = X*A(K)
            SUMA = SUMA + A(K)**2
            SUMB = SUMB + B(K)**2
            DOT = DOT + A(K)*B(K)
100   CONTINUE
      ANORM = SQRT(SUMA)
      BNORM = SQRT(SUMB)
```

(ب) جمع المصفوفات والضرب اللاموجه

تذكر أولا أن المصفوفة هى مجموعة متراصة ذات بعدين تكتب عناصرها عادة فى شكل مستطيل حيث يشير أول دليل إلى الصف فى المصفوفة والدليل الثانى إلى العمود فى المصفوفة .

افرض مصفوفتين $A = (a_{ij})$ و $B = (b_{ij})$ بهما عدد $m \times n$ من العناصر . وتكتب مجموع A و B بالصورة A + B ، وهى المصفوفة $m \times n$ التى عنصرها ij هو $a_{ij} + b_{ij}$ أى ، نحصل عليها بجمع العناصر المناظرة فى A و B . حاصل ضرب المصفوفة A والقيمة غير المتجهة k ، تكتب k.A وهى المصفوفة $m \times n$ التى عنصرها ij هو ka_{ij} أى ، نحصل عليها بضرب كل عنصر من A فى k .

وعلى سبيل المثال ، افرض المصفوفتين A و B 2×3 هما المصفوفتان 2×3 التاليتان :

$$A = \begin{pmatrix} 1 & -3 & 5 \\ 2 & 0 & -6 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4 & 1 & -2 \\ 1 & -5 & -3 \end{pmatrix}$$

وعلى ذلك :

$$A + B = \begin{pmatrix} 1+4 & -3+1 & 5-2 \\ 2+1 & 0-5 & -6-3 \end{pmatrix} = \begin{pmatrix} 5 & -2 & 3 \\ 3 & -5 & -9 \end{pmatrix}$$

$$-3A = \begin{pmatrix} -3\cdot 1 & -3\cdot(-3) & -3\cdot 5 \\ -3\cdot 2 & -3\cdot 0 & -3\cdot(-6) \end{pmatrix} = \begin{pmatrix} -3 & 9 & -15 \\ -6 & 0 & 18 \end{pmatrix}$$

وكما في الجبر العادي $2A - 4B = 2A + (-4B)$ ومن ثم :

$$2A - 4B = \begin{pmatrix} 2 & -6 & 10 \\ 4 & 0 & -12 \end{pmatrix} + \begin{pmatrix} -16 & -4 & 8 \\ -4 & 20 & 12 \end{pmatrix} = \begin{pmatrix} -14 & -10 & 18 \\ 0 & 20 & 0 \end{pmatrix}$$

افترض الآن A و B و C و D مجموعات متراصة في صورة مصفوفة كل منها $(M \times N)$ و X و Y كميات غير متجهة. فيما يلي جزء من برنامج الفورتران الذي يخزن مجموع A و B في C ويخزن X*A—Y*B في المصفوفة D :

```
      DO 100 L = 1, M
             DO 200 J = 1, N
                    C(I, J) = A(I, J) + B(I, J)
                    D(I, J) = X*A(I, J) + Y*B(I, J)
200          CONTINUE
100   CONTINUE
```

لاحظ أننا استخدمنا حلقات DO المتداخلة بسبب وجود دليلين .

(جـ) رمز التجميع

قبل أن نعرف ضرب المصفوفات ، سيكون من الأنسب أولا أن نقدم رمز التجميع Σ (الحرف اللاتيني سيجما (Sigma)).

افرض $f(k)$ هي التعبير الجبري المشتمل على المتغير k إذن فالتعبير :

$$\sum_{k=1}^{n} f(k) \quad \text{أو المكافئ} \quad \sum_{k=1}^{n} f(k)$$

له المعنى التالي : أولا نجعل $k = 1$ في $f(k)$ لنحصل على $f(1)$

$f(1)$

ثم نجعل $k = 2$ في $f(k)$ لنحصل على $f(2)$ ونضيف هذه القيمة على $f(1)$ لنحصل على :

$f(1) + f(2)$

بعد ذلك نجعل $k = 3$ في $f(k)$ لنحصل على $f(3)$ ونضيف هذه القيمة إلى المجموع السابق ، لنحصل على :

$f(1) + f(2) + f(3)$

ونستمر فى هذا الإجراء إلى أن نحصل على المجموع :

$$f(1) + f(2) + f(3) + \cdots + f(n-1) + f(n)$$

لاحظ أننا نزيد قيمة k بمقدار 1 عند كل خطوة حتى تصل قيمة k إلى n (ونشير إلى أنه فى استطاعتنا استخدام أى رمز آخر مثل i أو j بدلا من k) .

ونعمم أيضاً التعريف وذلك بالسماح للمجموع أن يتراوح ما بين أى رقم صحيح n_1 إلى أى رقم صحيح آخر n_2 حيث $n_1 \leq n_2$ أى نعرف :

$$\sum_{k=n_1}^{n_2} f(k) = f(n_1) + f(n_1+1) + f(n_1+2) + \cdots + f(n_2)$$

وعلى سبيل المثال :

لكل $f(k) = x_k$ يكون لدينا $\sum_{k=1}^{5} x_k = x_1 + x_2 + x_3 + x_4 + x_5$

ولكل $f(i) = a_i b_i$ يكون لدينا $\sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$

ولكل $f(j) = j^2$ يكون لدينا $\sum_{j=3}^{5} j^2 = 3^2 + 4^2 + 5^2 = 9 + 16 + 25 = 50$

ويكون لدينا أيضاً :

$$\sum_{k=1}^{5} (-1)^{k+1}(1/k) = 1 - 1/2 + 1/3 - 1/4 + 1/5$$

$$\sum_{k=1}^{p} a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + a_{i3} b_{3j} + \cdots + a_{ip} b_{pj}$$

$$\sum_{k=1}^{n+1} a_k x^{n+1-k} = a_1 x^n + a_2 x^{n-1} + \cdots + a_n x + a_{n+1}$$

(د) ضرب المصفوفات

افرض $A = (a_{ij})$ مصفوفة $m \times p$ و $B = (b_{ij})$ مصفوفة $p \times n$. حاصل ضرب A و B ، يكتب بالصورة AB ، وهو المصفوفة $C = (c_{ij})$ التى لها عدد $m \times n$ من العناصر والتى يحدد عنصرها c_{ij} بواسطة :

$$c_{ij} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{ip} b_{pj} = \sum_{k=1}^{p} a_{ik} b_{kj}$$

(يمكن أن نرى c_{ij} على أنها قيمة غير متجهة نحصل عليها بحاصل الضرب الداخلى (نقطة) للصف i فى A والعمود j فى B) .

على سبيل المثال ، افرض A و B هما المصفوفتان (2×3) و (3×4) التاليتان على الترتيب :

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad \text{و} \quad B = \begin{pmatrix} 5 & 6 & 7 & 8 \\ -1 & -2 & -3 & -4 \\ 0 & 9 & 0 & 9 \end{pmatrix}$$

لنحصل على حاصل الضرب $C = AB$ نذكر أن C ستكون مصفوفة (2×4) . العنصر c_{11} هو حاصل الضرب الداخلى الصف الأول فى A والعمود الأول فى B :

$$c_{11} = (1, 2, 3)\cdot(5, -1, 0) = 5 - 2 + 0 = 3$$

العنصر c_{12} هو حاصل الضرب الداخلى للصف الأول فى A والعمود الثانى فى B :

$$c_{12} = (1, 2, 3)\cdot(6, -2, 9) = 6 - 4 + 27 = 29$$

بالمثل :

$$c_{13} = 7 - 6 + 0 = 1$$
$$c_{14} = 8 - 8 + 27 = 27$$

الصف الثانى فى C نحصل عليه بضرب الصف الثانى فى A بكل الأعمدة فى B :

$$c_{21} = 20 - 5 + 0 = 15$$
$$c_{22} = 24 - 10 + 54 = 68$$
$$c_{23} = 28 - 15 + 0 = 13$$
$$c_{24} = 32 - 20 + 54 = 66$$

وبالتالى :

$$C = \begin{pmatrix} 3 & 29 & 1 & 27 \\ 15 & 68 & 13 & 66 \end{pmatrix}$$

لاحظ أن حاصل الضرب BA غير معروف حيث أن عدد الأعمدة فى A (4) ليس مساوياً لعدد الصفوف فى A (2) .

وبصورة عامة إذا كانت A مصفوفة $(M \times L)$ و B مصفوفة $(L \times N)$ فإن العنصر c_{IJ} فى مصفوفة حاصل الضرب C=AB يمكن أن يحسب كما يلى :

```
      C(I, J) = 0.0
      DO 100 INDEX = 1, L
            C(I, J) = C(I, J) + A(I, INDEX)*B(INDEX, J)
100   CONTINUE
```

ما يلى هو فى الواقع البرنامج الفرعى الكامل SUBROUTINE الذى يحسب C :

```
      SUBROUTINE MATMUL(A, B, M, L, N, C)
      DIMENSION A(M, L), B(L, N), C(M, N)
      DO 300 I = 1, M
            DO 200 J = 1, N
                  C(I, J) = 0.0
                  DO 100 INDEX = 1, L
                        C(I, J) = C(I, J) + A(I, INDEX)*B(INDEX, J)
100               CONTINUE
200         CONTINUE
300   CONTINUE
      RETURN
      END
```

٨ — ١٠ المعـــادلات الخطيـــة

نكرس هذا القسم لحل نظام مكون من عدد n من المعادلات الخطية فيها عدد n من المجاهيل . أولا ندرس الحالة الخاصة للنظام المثلثى . وبعد ذلك ندرس الحالة العامة باستخدام طريقة الحذف لجاوس .

(ا) النظام المثلثى

يقال أن النظام التالى من المعادلات الخطية فى الصيغة المثلثية

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots\cdots\cdots + a_{1,n-1}x_{n-1} + a_{1n}x_n &= b_1 \\
a_{22}x_2 + \cdots\cdots\cdots + a_{2,n-1}x_{n-1} + a_{2n}x_n &= b_2 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
a_{n-2,n-2}x_{n-2} + a_{n-2,n-1}x_{n-1} + a_{n-2,n}x_n &= b_{n-2} \\
a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\
a_{nn}x_n &= b_n
\end{aligned}$$

المجاهيل هنا هى x_1 ، x_2 ، ... x_n والمعاملات هى a_{11} ، a_{12}... a_{1n} ، ... a_{nn} . ونفرض أن النظام حل مفرد ؛ نظراً لعدم وجود أى حد من حدود القطر a_{11} ، a_{22} ، ... a_{nn} مساوياً للصفر . وسوف نشير هنا أيضاً إلى المعادلات بواسطة L_1 ، L_2 ، ... L_n على الترتيب .

ولإيجاد الحل x_n فى آخر معادلة L_n نحصل على :

$$x_n = b_n/a_{nn}$$

الآن ولإيجاد بقية المجاهيل نحل بالإجراء المعرف بالإحلال الخلفى . وبالتحديد ، نعوض بقيمة x_n فى المعادلة قبل الأخيرة L_{n-1} وذلك لإيجاد x_{n-1} فنحصل على :

$$x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}$$

بعد ذلك نعوض عن x_n و x_{n-1} فى المعادلة الثالثة قبل الأخيرة L_{n-2} ونحل لإيجاد x_{n-2} فنحصل على :

$$x_{n-2} = (b_{n-2} - a_{n-2,n-1}x_{n-1} - a_{n-2,n}x_n)/a_{n-2,n-2}$$

وهكذا ، أى ، لكل من $k < n$ نعوض عن x_n ، .... x_{k+1} فى L_k ونحل لإيجاد x_k فنحصل على :

$$x_k = (b_k - a_{k,k+1}x_{k+1} - \cdots - a_{kn}x_n)/a_{kk} = \left(b_k - \sum_{m=k+1}^{n} a_{km}x_m\right)\Big/ a_{kk}$$

(اسم الإحلال الخلفى يأتى من حقيقة أننا نحصل على x_k بالترتيب العكسى) .

افرض أن النظام المثلثى السابق تم تخزينه فى الحاسب بواسطة مجموعة متراصة $N \times (N + 1)$. يخزن البرنامج الصغير الفرعى BKSUB التالى الحل فى مجموعة متراصة X :

```
      SUBROUTINE BKSUB(A, N, L, X)
      DIMENSION A(N, L), X(N)
      X(N) = A(N, N + 1)/A(N, N)
      NN = N - 1
      DO 100 J = 1, NN
            K = N - J
C           CALCULATING SUM FOR X(K)
            SUM = 0.0
            NNN = N - K
            DO 200 JJ = 1, NNN
                  M = K + JJ
                  SUM = SUM + A(K, M)*X(M)
  200       CONTINUE
C           SOLVING FOR X(K)
            X(K) = (A(K, N + 1) - SUM)/A(K, K)
  100 CONTINUE
      RETURN
      END
```

لاحظ حلقات DO المتداخلة فى البرنامج الفرعى . تشير حلقة DO الخارجية إلى المتغير x_k فى المعادلة L_k وتشير حلقة DO الداخلية إلى التجميع فى الصيغة الرياضية x_k . يجب أن نأخذ فى الاعتبار أيضاً أننا نحصل على x_k بالترتيب العكسى وأن التجميع لـ x_k مبدأ من $k+1$ إلى n .

لاحظ أيضاً أن L تشير إلى عدد الأعمدة فى A ومن ثم ، يجب أن تساوى دائماً N + 1 .

(ب) النظام العام

النظام العام لعدد n من المعادلات الخطية E_1 ، E_2 ، $E_n \dots$ التى فيها عدد n من المجاهيل x_1 و x_2 و .... و x_n يمكن أن يشار إليه بما يلى :

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

سنعتبر أن النظام له حل فريد . لاحظ اننا نستطيع دائماً أن نبدل معادلتين أو نضيف مضاعفات معادلة ما إلى معادلة أخرى بدون أن نغير الحل . سنناقش الآن طريقة الحذف لجاوس والتى تستخدم هذه الخطوات السابقة لتحويل النظام إلى نظام مكافىء فى الصيغة المثلثية . وبمجرد أن يصبح فى الصيغة المثلثية ، نستخدم الإحلال الخلفى لإيجاد الحل .

أولا نحذف x_1 من المعادلات E_2 و E_3 و ... و E_n بالخطوتين التاليتين :

١ - أوجد المعادلة E_k بحيث يكون المعامل a_{k1} لـ x_1 أكبر قيمة مطلقة بين كل معاملات x_1 وليدل E_k مع E_1 . وبصورة أخرى نبدل المعادلات بحيث يكون المعامل a_{11} أكبر قيمة مطلقة بين معاملات x_1 (وهذا يضمن أن $a_{11} \neq 0$ ويؤدى أيضاً إلى دقة أفضل) .

٢ - استخدم المعادلة الأولى E_1 وبالتحديد ، المعامل a_{11} لـ x_1 لحذف x_1 من المعادلات الباقية كما يلى . لكل من $k > 1$ اضرب E_1 فى $(m_{k1} = -a_{k1}/a_{11})$ واجمع المعادلة إلى E_k (المعامل a_{11} يسمى المحور فى هذه الخطوة) .

الخطوات السابقة تحول النظام إلى الشكل التالى :

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

(ومن الطبيعى ، لا يستلزم ذلك أن تكون a_{ij} ، b_i فى النظام الجديد هى نفسها a_{ij} ، b_i فى النظام الأصلى) .

نكرر العملية السابقة مع النظام الفرعى E_2 و . . . و E_n أى ، نحذف x_2 من المعادلات E_3 و . . . . و E_n بطريقة مشابهة الخطوتين :

١ – بدل المعادلات (فيما عدا E_1) بحيث يكون العنصر المحورى a_{22} أكبر قيمة مطلقة بين معاملات x_2 فى النظام الفرعى .

٢ – اضرب E_2 فى $(m_{k2} = -a_{k2}/a_{22})$ واجمعها على E_k وذلك لكل $k > 2$.

وتحول هذه الخطوات النظام إلى الشكل التالى :

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \cdots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3' + a_{24}x_4 + \cdots + a_{2n}x_n &= b_2 \\
a_{33}x_3 + a_{34}x_4 + \cdots + a_{3n}x_n &= b_3 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
a_{n3}x_3 + a_{n4}x_4 + \cdots + a_{nn}x_n &= b_n
\end{aligned}
$$

بعد تكرار الاجراء السابق عدد $n - 1$ من المرات ، سيتحول النظام إلى الصيغة المثلثية التى يمكن أن تحل عندئذ بالإحلال الخلفى .

نريد الآن أن نترجم النظام الحسابى (الخوارزم) السابق إلى برنامج صغير فرعى SUBROUTINE يسمى GAUSS . سنعتبر أن نظام المعادلات الخطية مخزنة فى الذاكرة بمصفوفتها المتزايدة ، أى ، المجموعة المتراصة $(N \times (N+1))$

$$
\begin{pmatrix}
A_{11} & A_{12} & \ldots & A_{1N} & A_{1,N+1} \\
A_{21} & A_{22} & \ldots & A_{2N} & A_{2,N+1} \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
A_{N1} & A_{N2} & \ldots & A_{NN} & A_{N,N+1}
\end{pmatrix}
$$

حيث العمود $N + 1$ للمصفوفة هو عمود الثوابت للمعادلات . برنامجنا الفرعى GAUSS سيستخدم البرامج الثلاثة الفرعية الآتية المعطاة فى مسألة ٧ – ١٤ .

١ – SUBROUTINE FIND(A, N, L, K, J)

والذى يحدد الصف J بحيث يحتوى A(J, K) على القيمة العظمى المطلقةبين A(K, K) و A(K + 1, K) و . . . . . و A(N,K)

٢ – SUBROUTINE CHANGE(A, N, L, K, J)

والذى يبدل عناصر الصف K مع العناصر المناظرة فى الصف J فى A .

٣ SUBROUTINE ROWMUL(A, N, L, K, J, D)

والذى يجمع الصف K إلى الصف J عدد D من المرات .

(فى هذه البرامج الفرعية ، A مجموعة متراصة مصفوفة فى صورة N × L)

يمكن أن نحذف x_1 من المعادلات E_3 و .... و E_n باستخدام جزء البرنامج التالى .

```
      CALL FIND(A, N, N + 1, 1, J)
      CALL CHANGE(A, N, N + 1, 1, J)
      DO 100 I = 2, N
            CALL ROWMUL(A, N, N + 1, 1, I, -A(I, 1)/A(1, 1))
100   CONTINUE
```

بالمثل ، يمكن أن نحذف x_2 من المعادلات E_2 و .... و E_n باستخدام جزء البرنامج :

```
      CALL FIND(A, N, N + 1, 2, J)
      CALL CHANGE(A, N, N + 1, 2, J)
      DO 200 I = 3, N
            CALL ROWMUL(A, N, N + 1, 2, I, -A(I, 2)/A(2, 2))
200   CONTINUE
```

وهكذا . يستخدم البرنامج الفرعى عندما يكتمل حلقة DO لكل من الخطوات السابقة . ثم يستخدم البرنامج الفرعى BKSUB لإيجاد الحل بعد أن يكون النظام فى الصيغة المثلثية .

```
      SUBROUTINE GAUSS(A, N, L, X)
C
C     SOLUTION BY GAUSSIAN ELIMINATION
C
      DIMENSION A(N, L), X(N)
      NN = N - 1
      DO 99 K = 1, NN
            CALL FIND(A, N, N + 1, K, J)
            CALL CHANGE(A, N, N + 1, K, J)
            KK = K + 1
            DO 88 I = KK, N
                  CALL ROWMUL(A, N, N + 1, K, I, -A(I, K)/A(K, K))
88          CONTINUE
99    CONTINUE
      CALL BKSUB(A, N, N + 1, X)
      RETURN
      END
```

مسائل

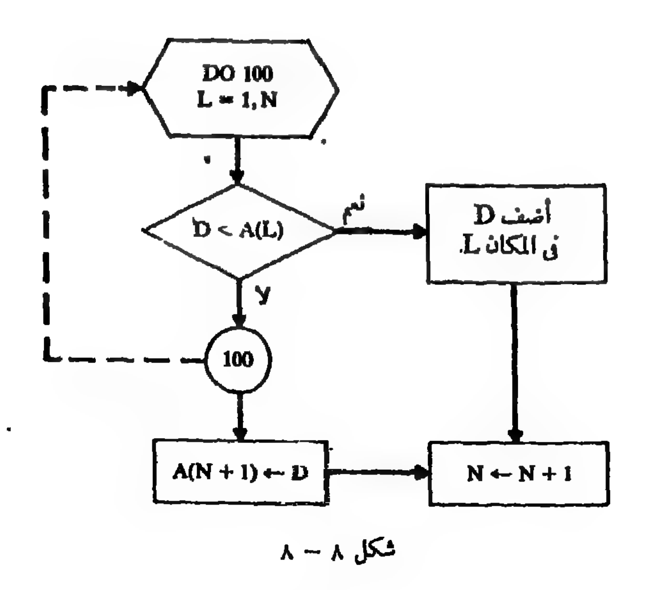
تقنية البرمجة الفنية

٨ — ١ عدل برنامج الفرز الفقاعى فى مسألة ٦ — ١٢ ليحتوى العداد الذى يعد عدد مرات التبديل ، إنه البرنامج إن لم يحدث تغيير فى أى عملية مسح (أو مرور) .

٨ — ٢ ترجم إلى الفورتران خريطة سير العمليات فى شكل ٨ — ٢ والتى يتم بها إدماج مجموعتين متراصتين A و B مفروزتين فى مجموعة متراصة C مفروزة اختبر البرنامج بالبيانات التالية :

A: 1, 5, 6, 12, 14, 21;   B: 2, 3, 9, 16, 18, 19, 24, 28

٨ – ٣ ترجم إلى الفورتران خريطة سير العمليات فى شكل ٨ – ٣ والتى تعطى النظام الحسابى (الخوارزم) للبحث الثنائى :



شكل ٨ – ٨

٨ – ٤ (تحديث) افرض أن A (1) و A (2) و ..... و A (N) مفروزه فى الترتيب التصاعدى ، وافرض أن D ليست فى المجموعة المتراصة A . تقوم خريطة سير العمليات فى شكل ٨ – ٨ بتحديث A باستخدام البحث الخطى لإيجاد المكان L حيث يجب أن تضاف D . ترجم خريطة سير العمليات إلى الفورتران .

٨ – ٥ (تحديث) عدل البرنامج فى مسألة ٨ – ٤ بحيث يستخدم البحث الثنائى لإيجاد المكان L حيث يجب أن تضاف D .

٨ – ٦ (بحث وتحديث) افرض أن A(1) و A(2) و .... و A(N) مفروزه فى الترتيب التصاعدى و D أى عنصر معطى . إذا كانت D تنتمى إلى المجموعة المتراصة A فنريد أن نجد مكانها ، لكن إذا كانت D لا تنتمى إلى المجموعة المتراصة A ، عندئذ نريد أن نقوم بتحديث A وذلك بإيجاد المكان L حيث يجب أن تضاف D .

(ا) اكتب مثل هذا البرنامج باستخدام البحث المتتالى .

(ب) اكتب مثل هذا البرنامج باستخدام البحث الثنائى .

٨ – ٧ (الفرز بالإضافة) . يسمى نظام حسابى (خوارزم) آخر للفرز بالفرز بالإضافة ويتم وصفه فيما يلى . الفكرة الأساسية هى إضافة عنصر فى مكانه المناسب بالنسبة لجزء مفروز فعلا من الملف . بالتحديد افرز العنصرين الأولين A(1) و A(2) أضف العنصر الثالث بحيث يكون A(1) و A(2) و A(3) مرتبة بطريقة مناسبة وبصورة عامة تضاف A(K) إلى جزء المجموعة المتراصة المفروز A(1) و ... و A(K – 1) بحيث تفرز A(1) و ... و A(K) بطريقة مناسبة .

اكتب برنامجاً للفرز بالإضافة . لاحظ أن المكان الذى يجب أن تضاف فيه A(K) يمكن أن يوجد بالبحث المتتالى أو البحث الثنائى . ناقش العدد الأقصى للمقارنات التى يمكن أن نحتاجها فى كل حالة . ناقش عيوب الفرز بالإضافة .

٨ – ٨ (فرز بالإدماج) — هناك نظام حسابى (خوارزم) آخر للفرز معروف جيدا وكفاءته عالية يسمى فرز بالإدماج والمخلوط

الأساسية للنظام الحسابى (الخوارزم) موضحة فيما يلى . فى المرور الأول ، افرز الزوج الأول والزوج الثانى والزوج الثالث وهكذا إلى أن نفرز كل الأزواج ، أى ، نفرز كلا من :

$$\{A_1, A_2\}, \{A_3, A_4\}, \{A_5, A_6\}, \{A_7, A_8\}, \{A_9, A_{10}\}, \ldots$$

فى المرور الثانى يدمج الزوج الأول من عنصرين من أجزاء المجموعة المتراصة المفروزة ثم يدمج الزوج الثانى من عنصرين من أجزاء المجموعة المتراصة المفروزة وهكذا . فى نهاية المرور الثانى يكون قد تم فرز كل من الرباعيات :

$$\{A_1, A_2, A_3, A_4\}, \{A_5, A_6, A_7, A_8\}, \{A_9, A_{10}, A_{11}, A_{12}\}, \ldots$$

تدمج بعد ذلك أزواج من الرباعيات فى المرور الثالث ، وهكذا إلى أن يتم فرز المجموعة المتراصة بأكملها . اكتب برنامجاً لتنفيذ هذه الفكرة ثم بين أن هذا النظام الحسابى (الخوارزم) يحتاج تقريباً إلى عدد $n \log_2 n$ من المقارنات .

٨ — ٩ (أ) اكتب برنامجاً صغيراً فرعياً DELETE لإلغاء جزء من مجموعة متراصة معطاة A(IN) و A(IN + 1) و . . . . A(LAST) من المجموعة المتراصة الأصلية A .

(تلميح . حرك جزء المجموعة المتراصة A(LAST + 1) و . . . . . . A(N) إلى أعلى عدد مناسب من الفجوات) .

(ب) افرض أن المجموعة المتراصة ID(1) و ID(2) و . . . . ID(N) مخزنة فى الذاكرة ولكن بعض المداخل أصفار . اكتب برنامجاً لضغط المجموعة المتراصة ID باستخدام DELETE

٨ — ١٠ افرض أن كل مندوب مبيعات فى متجر له رقم ID . تثقب بطاقة لكل صفقة يقوم بها مندوب البيع عليها ID وكمية البيع . فى نهاية اليوم تجمع هذه البطاقات ، ولكن بدون ترتيب . اكتب برنامجاً لحساب إجمالى البيع الذى تم بواسطة كل مندوب بيع .

٨ — ١١ افرض أن مركز الأنشطة الحاسبة فى جامعة يتقاضى من المستفيدين نتيجة خدماته 600.00 $ لكل ساعة . ويخصص لكل مستفيد ID لكل تشغيلة يتم إجراؤها على الحاسب . تثقب بطاقة برقم ID المستفيد والوقت الذى تطلبته التشغيلة بالثوانى . وتكون هذه البطاقات غير مفروزة . وبفرض إضافة تكاليف إدارية قيمتها 5.00$ لكل تشغيلة على الحاسب . اكتب برنامجاً لحساب الفاتورة الكلية لكل مستفيد . استخدم بطاقة خلفية لإنهاء البرنامج .

٨ — ١٢ فى الملف الرئيسى لشركة إدلر للخدمات التليفونية يتكون كل سجل من رقم حساب العميل ، وتكلفته أو تكلفتها الأساسية الشهرية . تثقب هذه السجلات على بطاقات كل سجل على بطاقة . ويفرز الملف الرئيسى تبعاً لرقم الحساب . عند حدوث مكالمة تليفونية للمسافات الطويلة ، تثقب بطاقة برقم حساب العميل وتكلفة المكالمة وهذه البطاقات تكون غير مفروزة بأى طريقة .

افرض أن فواتير التليفون تجهز فى نهاية كل شهر . باعتبار أن شركة أدلر للخدمات التليفونية لا تخدم أكثر من 100 عميل ، وأن بطاقات تكلفة المكالمات للمسافات الطويلة تفصل عن الملف الرئيسى ببطاقة خالية . وأن المجموعة تنتهى ببطاقة خلفية . اكتب برنامجاً لحساب الفواتير لكل عميل .

الحسابات العددية

٨ – ١٣ اكتب برنامجاً فرعياً لدالة HORNER(A, N, X) والتى تحسب قيمة كثيرة الحدود برتبة N − 1

$$A_1X^{N-1} + A_2X^{N-2} + \cdots + A_{N-1}X + A_N$$

٨ – ١٤ أوجد جذر الدالة $f(x) = x^3 - 4x^2 + 6x + 7$ ما بين $x = 2$ و $x = 3$. بترجمة خريطة سير العمليات الخاصة بتصنيف المدى فى شكل ٨ – ٥ إلى فورتران .

٨ – ١٥ حدد وأوجد الجذور الثلاثة الحقيقية للدالة $f(x) = x^3 + x^2 - 37x + 33$ ما بين $x = -10$ ، $x = 10$ باستخدام النظام الحسابى (الخوارزم) المبين فى شكل ٨ – ٦ وذلك لتحديد الجذور والنظام الحسابى (الخوارزم) بتصنيف المدى لأبعاد الجذور .

٨ – ١٦ أوجد جذرين حقيقيين للدالة $f(x) = x^2 - \sin x$

٨ – ١٧ أوجد المساحة تحت المنحنى $y = x^2 - 3x + 4$ ما بين $x = 3$ ، $x = 5$

٨ – ١٨ أوجد المساحة تحت المنحنى $y = x^2 + \sin x$ ما بين $x = 0$ ، $x = 2$

٨ – ١٩ اكتب برنامجاً فرعياً Subroutine NEWTON (A, N, X, Y, Z) يحسب قيمة $Y = F(X)$ ، $Z = F'(X)$ حيث

$$F(X) = A_1X^{N-1} + A_2X^{N-2} + \cdots + A_{N-1}X + A_N$$

و $F'(X)$ هى تفاضل $F(X)$. قارن النتائج مع المسألة ٨ – ١٣ . (تلميح : انظر المناقشة فى آخر قسم ٨ – ٦) .

٨ – ٢٠ (نيوتن – رابسون) ادرس المعادلة الدالية :

$$F(X) = 0$$

اجعل X_0 تشير إلى قيمة فى منطقة حل المعادلة . تكون المتتالية :

$$X_1 = X_0 - F(X_0)/F'(X_0), \quad X_2 = X_1 - F(X_1)/F'(X_1), \quad \ldots$$

أى ،

$$X_{N+1} = X_N - F(X_N)/F'(X_N)$$

(هنا تشير $F'(X)$ إلى تفاضل $F(X)$ تحت ظروف معينة ، تفاصيلها تقع خارج نطاق هذا المنهج ، المتتالية X_0 و X_1 و X_2 ... تؤول إلى جذر $F(X)$ (تسمى هذه العملية طريقة نيوتن – رابسون) اكتب برنامجاً باستخدام هذه الطريقة لحساب $\sqrt{5}$ ، أى ، لإيجاد حل :

$$X^2 - 5 = 0$$

اختر $X_0 = 3$ وانه البرنامج عندما تصبح $| X_{N+1} - X_N | \leq 0.001$

٨ – ٢١ استخدم طريقة نيوتن – رابسون لإيجاد جذور $x^2 + x - 11 = 0$ إلى ثلاث خانات عشرية . أولا اختر $x_0 = -5$ ثم $x_0 = 5$

المتجهات والمصفوفات والمعادلات الخطية

٨ – ٢٢ افرض أن $A = (2, -1, 0, -3)$ و $B = (1, -4, 4, 3)$ و $C = (1, 3, -2, 5)$

(أ) أوجد (١) $2A - 3B$ ، (٢) $5A - 3B - 4C$ ، (٣) $A.B$ و $A.C$ و $B.C$ ، (٤) $\|A\|$ و $\|B\|$ و $\|C\|$

(ب) اكتب برنامج فورتران لحساب ما سبق .

٨ – ٢٣ افرض

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 4 & 0 & -3 \\ -1 & -2 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 2 & -3 & 0 & 1 \\ 5 & -1 & -4 & 2 \\ -1 & 0 & 0 & 3 \end{pmatrix},$$

و

$$D = \begin{pmatrix} 2 & 2 \\ 3 & -1 \end{pmatrix}$$

أوجد :

(أ)	(g) D^3	(d) AC	(a) A + B
(ب)	(h) DA	(e) BC	(b) 3A − 4B
(جـ)	(i) DB	(f) D^2	(c) 2A + 3B

٨ – ٢٤ افرض ست بطاقات كل منها تحتوى على ثلاثة أرقام حقيقية . أكتب البرنامج الذى يقرأ هذه الأرقام فى صفوف المصفوفتان A و B حيث أن كل منها 3×3 ثم بعد ذلك يحسب ويطبع : (أ) $5A + 2B$ (ب) $3A - 7B$ (جـ) AB ، (د) BA ، (هـ) A^2

٨ – ٢٥ اعتبر المصفوفة (6×6) A بها كل عناصر القطر الرئيسى تساوى 0 وكل العناصر فوق القطر الرئيسى تساوى 1 وكل العناصر تحت القطر الرئيسى تساوى -1 كما يلى :

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 & 1 \\ -1 & -1 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 1 & 1 \\ -1 & -1 & -1 & -1 & 0 & 1 \\ -1 & -1 & -1 & -1 & -1 & 0 \end{pmatrix}$$

اكتب البرنامج الذى يخزن A فى الحاسب ثم يحسب ويطبع A و A^2 و A^3

٨ – ٢٦ أوجد حل النظام التالى :

$$\begin{aligned} 3x + 4y - 5z + 2s + 6t &= 10 \\ 5y + 7z - 4s - 6t &= -1 \\ 2z + 6s - 3t &= -35 \\ 8s + 3t &= -11 \\ 5t &= 35 \end{aligned}$$

٨ – ٢٧ أوجد حل النظام التالى :

$$\begin{aligned} 2x + 3y + 4z - 5s + 7t &= -35 \\ 8x - 2y - 3z + 9s + 3t &= 53 \\ 4y + 6z - 3s - 2t &= -33 \\ 5x - 7y + 8z + 3s - 9t &= -19 \\ 3x + 5y - 2z + 4s + 6t &= 27 \end{aligned}$$

٨ – ٢٨ (جاوس – جوردان) افرض نظام به عدد N من المعادلات فى N من المجاهيل مخزنة فى الحاسب فى مجموعة متراصة A على صورة المصفوفة $N \times (N+1)$ يمكن أن نعدل طريقة جاوس وتحول المصفوفة إلى الشكل :

$$\begin{pmatrix} A_{11} & 0 & \dots & 0 & A_{1,N+1} \\ 0 & A_{22} & \dots & 0 & A_{2,N+1} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & A_{NN} & A_{N,N+1} \end{pmatrix}$$

بدلا من الصيغة المثلثية وعلى ذلك يكون الحل :

$$X_1 = A_{1,N+1}/A_{11}, \quad X_2 = A_{2,N+1}/A_{22}, \quad \dots, \quad X_N = A_{N,N+1}/A_{NN}$$

(ويسمى هذا الإجراء طريقة جاوس – جوردان) . اكتب برنامجاً لتنفيذ هذا النظام الحسابى (الخوارزم) واختبر البرنامج بالنظام المعروض فى المسألة ٨ – ٢٤ .

٨ – ٢٩ اجعل مصفوفة مربعة $A\ (N \times N)$. وتسمى المصفوفة $B\ (N \times N)$ مقلوب A إذا كانت $AB = BA = I$ حيث I هى مصفوفة الوحدة أى المصفوفة التى تشتمل على آحاد على القطر الرئيسى وأصفار فى الأماكن الأخرى . إذا وجد مقلوب A فهو فريد ويرمز له بالرمز A^{-1} وفيما يلى نظام حسابى (خوارزم) لإيجاد A^{-1} أولا كون المصفوفة $(N \times 2N)$ بالصورة التالية :

$$(A, I) = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} & 1 & 0 & \dots & 0 \\ A_{21} & A_{22} & \dots & A_{2N} & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ A_{N1} & A_{N2} & \dots & A_{NN} & 0 & 0 & \dots & 1 \end{pmatrix}$$

أى النصف الأيسر هو المصفوفة A والنصف الأيمن هو I . باستخدام تعديل لطريقة جاوس ، تحول المصفوفة العليا إلى المصفوفة كما فى الشكل التالى :

$$(I, B) = \begin{pmatrix} 1 & 0 & \dots & 0 & B_{11} & B_{12} & \dots & B_{1N} \\ 0 & 1 & \dots & 0 & B_{21} & B_{22} & \dots & B_{2N} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & B_{N1} & B_{N2} & \dots & B_{NN} \end{pmatrix}$$

أى ، حيث النصف الأيسر الآن هو I والنصف الأيمن B هو مقلوب A . اكتب برنامجاً لتنفيذ هذا النظام الحسابى (الخوارزم) .

حلول المسائل المختارة

٨ – ٢٢

(١) $(1, 10, 12, -15)$ (٢) $(3, -5, 20, -44)$ (٣) $-3, -16, 12$ (٤) $\sqrt{14}, \sqrt{42}$, and $\sqrt{39}$

٨ – ٢٣

(أ) $\begin{pmatrix} 5 & -1 & -1 \\ -1 & 1 & 7 \end{pmatrix}$

(ب) $\begin{pmatrix} -13 & -3 & 18 \\ 4 & 17 & 0 \end{pmatrix}$

(جـ) $\begin{pmatrix} 14 & -2 & -5 \\ -3 & 0 & 17 \end{pmatrix}$

(د) $\begin{pmatrix} -5 & -2 & 4 & 5 \\ 11 & -3 & -12 & 18 \end{pmatrix}$

(هـ) $\begin{pmatrix} 11 & -12 & 0 & -5 \\ -15 & 5 & 8 & 4 \end{pmatrix}$

(و) $\begin{pmatrix} 10 & 2 \\ 3 & 7 \end{pmatrix}$

(ز) $\begin{pmatrix} 26 & 18 \\ 27 & -1 \end{pmatrix}$

(ح) $\begin{pmatrix} 2 & 4 & 12 \\ 3 & -6 & 2 \end{pmatrix}$

(ط) $\begin{pmatrix} 6 & -4 & 0 \\ 13 & 2 & -12 \end{pmatrix}$

٨ – ٢٦ $x = 3, y = -2, z = 5, s = -4, t = 7$

٨ – ٢٧ $x = 2, y = 1, z = -5, s = 3, t = -1$

الفصل التاسع

معلومات الحروف ، المتغيرات المنطقية والعمليات الحسابية

٩ ــ ١ مقدمــــة

تم تطوير الفورتران أساساً لترجمة المعادلات FORmula TRANslation . ومع ذلك ، يمكن أن يتناول الفورتران أيضاً معلومات حرفية . نذكر أنه يمكننا أن نطبع رسائل بوضع الرسالة بين زوج من الفصلات العليا فى جملة FORMAT . على سبيل المثال :

```
      WRITE(6, 10)
   10 FORMAT(1X, 'THE END')
```

ستسبب طبع الرسالة :

THE END

هذه الرسالة بها سبعة حروف T و H و E ومسافة خالية (التى نرمز إليها بواسطة b) ثم E وN و D . يسمح لنا الفورتران أيضاً بقراءة وتخزين مثل هذه السلسلة من الحروف والتعامل معها ، مثلا ، كعملية فرز قائمة أسماء أبجدية . سوف نناقش فى هذا الفصل معالجة الرموز باستخدام الفورتران .

وسوف نناقش أيضاً فى هذا الفصل . استخدام الفورتران لقيام بعمليات منطقية وإيجاد قيمة متغيرات منطقية .

٩ ــ ٢ تخزين الحروف

عندما تخزن الحروف فى ذاكرة الحاسب ، يتم تخزينها بأكواد . وفى الآلة الثنائية تخزن الحروف كسلسلة من 0's و 1's (أى اصفار وآحاد) . وواضح أنه يمكن تفسير مثل هذه السلسلة كرقم ثنائى . يطلق على هذا الرقم المكافىء العددى للمعلومات الحرفية . نقطة واحدة تحتاج إلى تأكيد وهى أن التمثيل الداخلى لـ .

1234

عند تخزينها كسلسلة حرفية يختلف تماماً فى الشكل عن تمثيلها الداخلى كعدد صحيح .

هناك حد أعلى لعدد الحروف التى يمكن أن تخزن فى أى خلية ذاكرة محددة . سيسمى هذا الرقم ،(والذى يتغير من آلة إلى أخرى) ، سعة الحرف فى الآلة (أو ببساطة سعة) وسيرمز له بالحرف M . وتكون M=4 لآلات سلسلة IBM 360/370 ، وتكون M=10 فى سلسلة CDC 6000 (ولجعل البرامج قابلة النقل من آلة إلى أخرى ننصح بأن نفترض M = 4) .

يمكن أن نستخدم أماكن صحيحة فقط إذ أنها تتجنب أى أخطاء تقريب قد تظهر عند تداول المعلومات الحرفية .

جملة DATA غير المنفذة ، هى إحدى الطرق لتخزين المعلومات الحرفية (نناقش جملة DATA بالتفصيل فى قسم ١٠-٢) . وعلى سبيل المثال ، الجملة :

```
      DATA NAME/'PAT'/
```

والتي يجب وضعها قبل أى جملة منفذة فى البرنامج ، ستقوم بتخزين السلسلة الحرفية PAT فى المكان المسمى NAME . وتعطى الفصلات العليا إشارة للحاسب أن PAT ستخزن كحروف ، وتستخدم الشرطات المائلة لتحيط عنصر الإدخال ، وبذلك تفصلها عن اسم المتغير .

وباعتبار أن سعة الحروف M = 4 فسوف تخزن PAT فى الذاكرة كالتالى :

NAME	P	A	T	b

أى أن السلسلة الحرفية ستخزن مضبطة من الطرف الأيسر فى مكان الذاكرة وتنتهى بمسافات خالية على اليمين .

من ناحية أخرى ، إفرض أنك أعطيت الجملة :

DATA ID/'JOHNSON'/

أى ، سوف تخزن السلسلة الحرفية JOHNSON فى مكان الذاكرة المسمى ID مضبطة من الطرف الأيسر . ولما كانت السعة M = 4 فسوف تبتر الحروف الزائدة من على اليمين ، ويتم تخزين JOHN فقط فى ID .

ID	J	O	H	N

بالتحديد ، ستخزن الحروف الأربعة الأولى فقط (الحروف الأربعة من أقصى اليسار) حيث M = 4 ويجدر ملاحظة التعليقات الآتية :

١ – بعض الحاسبات تسمح لظهور الثوابت الحرفية فى جمل منفذة . فى هذه الحالة يمكن أن تخزن المعلومات الحرفية بعبارة تخصيص .

على سبيل المثال ، الجملة :

NAME = 'PAT'

سيكون لها نفس تأثير الجملة

DATA NAME/'PAT'/

أى أننا سنخزن السلسلة الحرفية PAT بين الفصلات العليا فى موقع ذاكرة يسمى NAME .

٢ – مع M = 4 سوف تتطلب مرتين على الأقل لتخزين سلسلة الحروف JOHNSON على سبيل المثال ، أى ما يأتى :

DATA IDA/'JOHN'/
DATA IDB/'SON'/

DATA IDA/'JOHN'/, IDB/'SON'/

أو

DATA IDA, IDB/'JOHN', 'SON'/

والنتيجة أن JOHN ستخزن فى IDA وتخزن SON فى IDB :

IDA | J | O | H | N | IDB | S | O | N | b |

٣ – نستطيع أن نستخدم حقل H أيضاً فى جملة DATA بدلا من الفصلات العليا . على سبيل المثال :

DATA NAME/3HPAT/

تخزن أيضاً سلسلة الحروف PAT فى NAME (أنظر قسم ٩ ـ ٥) .

٤ – رغم أن بعض الحاسبات لاتسمح الثوابت الحرفية أن تظهر فى الجمل القابلة للتنفيذ كما فى (١) فإنه يمكننا دائماً نسخ محتويات مكان إلى داخل مكان آخر بجملة تخصيص . وعلى سبيل المثال وباعتبار M = 4 فسوف يقوم جزء البرنامج :

```
DATA NAME/'PAT'/
NEXT = NAME
```

بتخزين PAT_b وهى محتويات NAME فى مكان الذاكرة المسمى NEXT .

NEXT | P | A | T | b |

لاحظ أن NEXT و NAME كلاهما متغير ثابت . ننصح بعدم مزج الأنواع فى جمل التخصيص هذه .

تحذير : عندما يتجاوز طول الثابت الحرفى *l* فى جملة DATA سعة الحرف M تعطى بعض المترجمات إشارة خطأ . هناك احتمالات أخرى سوف يتم مناقشتها فى قسم (١٠ – ٢) ولكن كل المترجمات تعطى نفس النتيجة عندما يكون $l \leq M$.

٩ ـ ٣ حقـــل A

طريقة أخرى لتخزين المعلومات الحرفية هى قراءة المعلومات بواسطة جملة READ مثلا . عندما تقرأ المعلومات الحرفية أو تطبع ، فإننا نستخدم حقل A التى تأخذ مواصفاته الشكل :

Aw

حيث *w* هى عرض الحقل . (ونستطيع أن نستخدم *rAw* أيضاً ، حيث *r* هى معامل تكرار) .

(أ) إدخـــال

إفرض أن السعة M = 4 وتم تثقيب بطاقات بيانات كما يلى :

```
                   1                   2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
LONGFELLOW
```

وعلاوة على ذلك ، إفرض أمر الإدخال

```
   READ(5, 10) NAME
10 FORMAT(A3)
```

بهذا تخبر الحاسب أن سلسلة الحروف التى فى الأعمدة من 1 إلى 3 ستخزن فى المكان المسمى NAME وهنا عرض الحقل $w = 3$ لايتعدى سعة الحرف M ، ولذا ستخزن سلسلة الحروف مضبطة من الطرف الأيسر فى الذاكرة ومزودة بمسافه خالية على اليمين :

NAME	L	O	N	b

من ناحية أخرى ، إفرض أمر الإدخال هو :

```
   READ(5, 20) NAME
20 FORMAT(A10)
```

هنا يتجاوز عرض الحقل ($w = 10$) السعة M . وعلى ذلك سيتم تخزين الأحرف الأربعة فقط ($M = 4$) من الحقل والموجوده على أقصى اليمين فى الذاكرة :

NAME	L	L	O	W

لاحظ أن جملة DATA :

```
DATA NAME/'LONGFELLOW'/
```

سيكون لها نتيجة مختلفة . بالتحديد ، ستأمر جملة DATA بإدخال الحروف الأربعة $M = 4$ التى على أقصى اليسار فقط ويتم تخزينها فى الذاكرة :

NAME	L	O	N	G

يمكن تلخيص النتائج السابقة كما يلى :

قواعد إدخال الحروف باستخدام حقل A . إفرض M هى سعة الحرف و w هى عرض الحقل المعطى فى جملة FORMAT للإدخال.

١ – إذا كانت $M \geq w$ تخزن الحروف w من الحقل فى الذاكرة مضبطة من جهة اليسار مع إضافة مسافات خالية بطول $M - w$ على اليمين .

٢ – إذا كانت $M < w$ تخزن الحروف M التى على أقصى اليمين من الحقل فى الذاكرة .

وتبعاً لذلك فعند قراءة الإسم LONGFELLOW باستخدام بطاقة البيانات السابقة ومع $M = 4$ نحتاج على الأقل إلى ثلاثة أماكن ذاكرة للتخزين ، فمثلا :

```
   READ(5, 10) L1, L2, L3
10 FORMAT(3A4)
```

ستخزن أول أربع حروف LONG فى L1 والحروف الأربعة الثانية FELL فى L2 والحروف الأربعة الثالثة OW_{bb} فى L3 :

L1	L	O	N	G	L2	F	E	L	L	L3	O	W	b	b

فى حالة تغيير جملة FORMAT إلى :

```
10 FORMAT(2A4, A2)
```

تقرأ عشرة أعمدة فقط . ومع ذلك ستكون النتيجة النهائية مطابقة لأن الحروف OW تخزن مضبطة من اليسار فى L3 مع إضافة مسافتين خاليتين على اليمين .

من الواضح أننا نستطيع أن نستخدم أيضاً مجموعة متراصة لتخزين الأسم LONGFELLOW كما يلى :

```
      DIMENSION L(3)
      READ(5, 10) L
10    FORMAT(3A4)
```

أخيراً نؤكد أن المسافات تفهم كحروف خالية حينما تقرأ باستخدام حقل حرفى A مع أنها تفهم كأصفار باستخدام حقول رقمية .

(ب) الخــرج

نؤكد أولا أنه إذا احتوى مكان ذاكرة على معلومات حرفية فيجب أن يحتوى وبكل دقة على عدد M من الحروف (أى ، تضاف مسافات خالية على اليسار إذا كان عدد حروف سلسلة الحروف الأصلية أقل من M) . وقبل أن نذكر بصفة رسمية القواعد التى تستخدم مع حقل A فى الخرج سوف نعطى المثال التالى :

إفرض السعة M = 4 ويحتوى المكان NAME على :

NAME | P | A | T | b |

إذن سوف تطبع

```
      WRITE(6, 20) NAME
20    FORMAT(1X, A4)
```

محتويات NAME وهى .

PAT$_b$

هنا يكون عرض الحقل المعطى w هو نفسه السعة M = 4 ومع ذلك إذا كانت جملة FORMAT هى :

```
20    FORMAT(1X, 2A)
```

حيث عرض الحقل المعطى $w = 2$ أقل من M = 4 فسوف يطبع فقط الحرفان اللذان على أقصى اليسار فى المخزن ، وبالتحديد ستطبع :

PA

فقط .

من ناحية أخرى ، إذا كانت جملة FORMAT هى :

```
20    FORMAT(1X, A7)
```

حيث عرض الحقل المعطى $w = 7$ أكبر من M = 4 إذن ستطبع الحروف M = 4 من المخزن مضبطة من اليمين فى الحقل كما يلى :

$_{bbb}$PAT$_b$

بمعنى آخر ، يتكون الخرج من عدد ثلاث مسافات خالية $w - M = 3$ متبوعة بمحتويات NAME .

النتائج السابقة يمكن تلخيصها فيما يلى :

قواعد إخراج حقول حرفية باستخدام حقل A . إفرض M هى سعة الحرف و w هى عرض الحقل المعطى فى جملة FORMAT فى الخرج.

١ – إذا كانت $w \leq M$ تكون نتيجة الخرج عدد w من الحروف الموجودة أقصى اليسار فى المخزن ، أى سيبتر عدد $M - w$ من الحروف أقصى اليمين .

٢ – إذا كانت $w > M$ ، تكون نتيجة الخرج عدد $w - M$ من المسافات الخالية متبوعة بعدد M من الحروف التى بالمخزن .

مثال ٩ – ١

نريد أن نثقب بطاقات لـ 200 اسم ثم تخزن وتطبع بحيث يبدو الخرج كما يلى :

```
BROWN        ERIC
EVERLING     MICHAEL
LIPSON       AUDREY
........     .......
```

بفرض أنه لا يوجد إسم (أول أو أخير) يتجاوز 15 حرفاً ، لذلك نثقب الإسم الأخير مضبطاً من اليسار فى الأعمدة من 1 إلى 15 والإسم الأول مضبطاً من اليسار فى الأعمدة من 17 إلى 31 . (وعلى ذلك يجب أن نفصل الإسم الأخير عن الإسم الأول بمسافة خالية واحدة على الأقل) .

بفرض أن السعة M = 4 فكل اسم يحتاج إلى ثمانية أماكن ذاكرة على الأقل لتخزين الـ 31 حرفاً المحتملة أن يتكون منها الاسم . وتبعاً لذلك فإننا نستخدم مجموعة متراصة ذات بعدين (8 × 200) تسمى NAME . يجب أن نستخدم حلقات DO ضمنية من أجل تخزين وطباعة المتراصة صفا بصف وفيما يلى جزء البرنامج :

```
      DIMENSION NAME(200, 8)
      DO 99 K = 1, 200
            READ(5, 15) (NAME(K, I), I = 1, 8)
15          FORMAT(8A4)
            WRITE(6, 25) (NAME(K, I), I = 1, 8)
25          FORMAT(1X, 8A4)
99    CONTINUE
```

٤ – مناولة المعلومات الحرفية

كما سبق ذكره فى قسم ٩ – ٢ ، فإن محتويات أى مكان ذاكرة هى متسلسلة من 0's و 1's (بافتراض أن لدينا آلة ثنائية) من ثم يمكن أن نقوم بالعمليات المترابطة أيضاً على المكافىء العددى لسلاسل الحروف . وبذلك يمكن أن نتداول المعلومات الحرفية كما نرى من المناقشة التالية .

(أ) الفرز :

أحياناً كثيرة ، نريد أن نفرز وحدات بأسمائها الحرفية ، أى نريد عمل قائمة بالأسماء الأبجدية . لترتيب السلاسل الحرفية أبجدياً ، يجب أن نعرف أولا كيف تكود الحروف الأبجدية داخلياً . ولحسن الحظ ، فإن معظم المترجمات تكود الحروف الأبجدية فعلا بالترتيب تصاعدى . أى أن ، المكافىء الرقمى لحرف A أقل من المكافىء الرقمى B . كمثال ، إفرض أن IA و IB يحتويان على المعلومات الحرفية التالية :

IA | A | L | A | N | IB | D | A | L | E |

إذن سيكون المكافىء الرقمى لـ ALAN المخزن فى IA أقل من المكافىء الرقمى لـ DALE المخزن فى IB وتبعاً لذلك فالتعبير المترابط :

IA.LT.IB

يكون صحيحا . وبذلك نستطيع أن نستخدم الطرق المتعددة لفرز القيم الرقمية أيضاً عند فرز السلاسل الحرفية . وتستخدم المسألة ٩ – ٨ طريقة الفرز الفقاعى لفرز قائمة من الأسماء .

(ب) تحليـــل النص

إفرض جملة S تم تثقيبها فى بطاقة بيانات (تذكر أن بطاقة البيانات بها 80 عموداً) إدرس كيفية كتابة البرنامج الذى :

١ – يطبع الجملة S .

٢ – يحصى عدد المرات التى ظهر فيها الحرف E فى الجملة S .

إذا أردنا أن نطبع الجملة S فقط يمكن أن نخزن أربعة حروف فى مكان ذاكرة باستخدام مجموعة متراصة خطية بها 20 عنصراً كالآتى :

```
    DIMENSION L(20)
    READ(5, 15) L
15  FORMAT(20A4)
    WRITE(6, 25) L
25  FORMAT(1X, 20A4)
```

ومع ذلك لا نستطيع أن نحصى عدد المرات التى ظهر فيها الحرف E بواسطة هذا البرنامج ولكى نفعل ذلك يجب أن نخزن حرفاً واحداً فى كل مكان باستخدام مجموعة متراصة خطية بها 80 عنصر كما فى البرنامج التالى :

```
    DIMENSION L(80)
    DATA JJ/'E'/
    READ(5, 15) L
15  FORMAT(80A1)
    WRITE(6, 25) L
25  FORMAT(1X, 80A1)
    N = 0
    DO 99 K = 1, 80
        IF(L(K).EQ.JJ) N = N + 1
99  CONTINUE
    WRITE(6, 35) N
35  FORMAT(1X, 'THE LETTER E APPEARS', 1X, I2, 1X, 'TIMES')
```

لاحظ أننا نستخدم جملة DATA لتخزين الحرف E فى مكان يسمى JJ ثم نستخدم حلقة DO لمقارنة JJ مع كل حرف فى L أى مع كل حرف فى الجملة S .

٩ – ٥ حقــل – H

تذكر من قسم ٣–٨ أننا نستطيع أن نطبع رسائل فى المخرج الخاص بنا بإحاطة سلسلة الحروف المطلوبة فصلات عليا فى جملة الـ FORMAT الخاصة بالإخراج . على سبيل المثال ، إفرض أن X تحتوى على 123.45 إذن فزوج الجمل WRITE-FORMAT التالى .

```
    WRITE(6, 10) X
10  FORMAT(1X, 'THE BALANCE IS $', F8.2)
```

سينتج :

THE$_b$BALANCE$_b$IS$_b$$$_{bb}$123.45

بالمثل :

```
    WRITE(6, 20)
20  FORMAT('1', 'ACCOUNT NUMBER', 10X, 'BALA^ ^E')
```

سيطبع العنوان التالى على قمة صفحة جديدة كما يلى :

ACCOUNT NUMBER$_{bbbbbbbbbb}$BALANCE

نستطيع أيضاً أن نطبع رسائل باستخدام حقل H (حقل Hollerith) بالمواصفات التالية :

wH متبوعة بعدد *w* من الحروف المراد طباعتها .

يمكن تثقيب جمل FORMAT السابقة بالصورة المكافئة التالية :

```
10  FORMAT(1X, 16HTHE BALANCE IS $, F8.2)
20  FORMAT(1H1, 14HACCOUNT NUMBER, 10X, 7HBALANCE)
```

وهنا نؤكد أن عدد الحروف *w* لاتحاط بفصلات عليا (إلا إذا كان المطلوب طباعة الفصلات نفسها). نذكر فيما يلى مواصفات حقل H — ومواصفات الحقل الحرفى المكافىء :

حقل H	حقل الحروف
8HSEQUENCE	'SEQUENCE'
7HTHE END	'THE END'
15HNOW IS THE TIME	'NOW IS THE TIME'

ويعيب مواصفات حقل H أن المسافات يجب أن تعد بدقة . ويجدر أن نلاحظ أن المسافات بداخل حقل H لاتهمل ، حيث أن المسافة تحسب كحرف .

يمكن أيضاً استخدام حقل H للإدخال فنستطيع أن نستخدم حقل H لتحديد سلسلة حروف فى جملة DATA . فمثلا :

```
DATA I/3HPAT/, J/4HMARK/
```

لها نفس تأثير :

```
DATA I/'PAT'/, J/'MARK'/
```

نستطيع أيضاً أن نستخدم حقل H فى جملة FORMAT للإدخال فى مثل هذه الحالة يجب أن تظهر حروف زائفة (أو : مسافات خالية فى جملة FORMAT . وتوضح هذه الحالة فى المثال التالى :

مثل ٩ — ٢

إفرض أن أول بطاقتين للبيانات تم تثقيبها كالتالى :

```
                    1                   2                   3                   4
           123456789012345678901234567890123456789012345
البطاقة الأولى       ANDERSON       JOHN
البطاقة الثانية      BERGER         AUDREY
```

ويتم تنفيذ جزء البرنامج التالى :

```
      DO 100 K = 1, 2
         READ(5, 10)
10       FORMAT(5X, 15HAAAAABBBBBCCCCC, 15HXXXXXYYYYYZZZZZ)
         WRITE(6, 10)
100   CONTINUE
```

عندما تنفذ جملة READ أول مرة ، تستبدل السلاسل الحرفية فى الأعمدة من 6 إلى 20 والأعمدة من 21 إلى 35 بالحروف المذكورة فى مواصفات حقول H فى جملة FORMAT ، أى تتحول جملة FORMAT فعليا إلى :

```
10   FORMAT(5X, 15HANDERSON      , 15HJOHN           )
```

من ثم عندما تنفذ جملة WRITE يطبع الإسم الموجود على بطاقة البيانات . بالمثل عندما تنفذ جملة READ المرة الثانية ، يقرأ الاسم على بطاقة البيانات الثانية من خلال جملة FORMAT وسوف تطبع عندما تنفذ جملة WRITE وتبعاً لذلك سيبدو الخرج كما يلى :

```
ANDERSON       JOHN
BERGER         AUDREY
```

٩ ـ ٦ ثوابت منطقية ومتغيرات منطقية

خلال مناقشة جملة IF المنطقية فى الفصل الرابع قدمنا التعبيرات المترابطة التى هى أبسط شكل للتعبيرات المنطقية . والتعبيرات المترابطة هى تعبيرات تربط تعبيرين حسابين بإحدى أدوات الترابط الست التالية :

.EQ. .NE. .LT. .LE. .GT. .GE.

وتسمى التعبيرات المترابطة أيضاً التعبيرات المنطقية لأن كلا منها تعتبر قيمة منطقية TRUE أو FALSE ويمثل هذان الثابتان المنطقيان فى الفورتران بالتالى :

.TRUE. و .FALSE.

على الترتيب . ونؤكد أن الكلمات TRUE و FALSE يجب أن يسبق ويتبع كل منها نقطة .

تسمى أماكن التخزين (المتغيرات) التى تستخدم لتخزين ثوابت منطقية متغيرات منطقية . ومثل كل المتغيرات يتكون المتغير المنطقى من واحد إلى ستة حروف أبجدية رقية ويجب أن يكون أول حرف أبجدى كما يجب أن تعلن أسماء المتغيرات المنطقية بجملة النوع LOGICAL فمثلا الجملة :

LOGICAL I, A45X, CAR

تعلن أن I و A45X و CAR متغيرات منطقية . يمكن أيضاً أن نعلن عن مجموعات متراصة منطقية بجملة LOGICAL فمثلا :

LOGICAL A(15)

لاتعلن فقط أن A مجموعة متراصة منطقية ، ولكنها تحجز أيضاً 15 مكاناً فى الذاكرة باسم A . وحيث أن جملة LOGICAL غير منفذة ، لذا يجب أن تظهر قبل أى جملة منفذة فى البرنامج .

٩ ـ ٧ المعاملات والتعبيرات المنطقية

نستطيع أن نبنى التعبيرات المنطقية الأكثر تعقيداً من التعبيرات المترابطة فى الفورتران بواسطة المعاملات الثلاث الآتية (الموصلات) :

.AND.
.OR.
.NOT.

ومرة ثانية نؤكد على أن الكلمات AND و OR و NOT تسبقها وتتبعها نقطة .

إذا أعطيت أى تعبيرين منطقيين lexp1 و lexp2، فمن الممكن أن نكون التعبير المنطقى المركب التالى :

lexp1.AND.lexp2
lexp1.OR.lexp2
.NOT.lexp1

تعتمد القيم المنطقية لهذه التعبيرات المركبة على القيم المنطقية لكل من lexp 1 و lexp 2 وهى تظهر فى الجداول ٩ ـ ١ و ٩ ـ ٢ (حيث T تعنى تتحقق و F لايتحقق) . لاحظ أن :

١ ـ المعامل . AND . يتحقق عندما يتحقق lexp 1 و lexp 2 معاً .

٢ ـ المعامل . OR . لايتحقق عندما لايتحقق lexp 1 و lexp 2 معاً .

٣ ـ المعامل . NOT . هى أداة أحادية وهى تغير القيمة المنطقية لأى تعبير منطقى من T إلى F أومن F إلى T .

هذه المعاملات مثلها مثل الروابط المستخدمة فى اللغة الإنجليزية لتكوين الجمل المركبة .

جدول ٩ ـ ١

lexp1	lexp2	lexp1.AND.lexp2	lexp1.OR.lexp2
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	F

جدول ٩ ـ ٢

lexp1	.NOT.lexp1
T	F
F	T

مثال ٩ ـ ٣

(أ) ادرس التعبيرات المنطقية الآتية :

(١) (I.LT.J).OR.(K.GT.10)
(٢) (X.GT.1.0).AND.(K.LT.(I+10))
(٣) .NOT.(X.GT.0.0)

ويصبح لدينــا التالى :

يتحقق (١) إذا كانت $I < J$ أو $K > 10$.

يتحقق (٢) إذا كان كل من $X > 1$ و $K < I + 10$.

يتحقق (٣) إذا لم تتحقق $X > 0$ أى إذا كانت $X \leq 0$.

(ب) يقع الطلبة المبتدئون فى غلطة شائعة وهى ترجمة $1 \leq x \leq 2$ إلى الفورتران بكتابة :

1.0.LE.X.LE.2.0

يجب أن ندرك أن $1 \leq x \leq 2$ تعنى $1 \leq x$ و $x \leq 2$. ومن ثم الفورتران الصواب المكافىء هو :

(1.0.LE.X).AND.(X.LE.2.0)

رغم أن التعبيرات المنطقية تستخدم كثيراً فى جمل IF المنطقية ، يمكن أيضاً أن تدمج وتحل مسائل الجبر البوول والمنطق الافتراضى باستخدام هذه التعبيرات .

٩ — ٨ تخصيص القيم المنطقيــة

يمكننا أن نخزن ثوابت منطقية فى متغيرات منطقية بجملة تخصيص . فمثلا :

```
LOGICAL A
A = .TRUE.
```

تخصص الثابت .TRUE. إلى مكان الذاكرة المسمى A . عموماً ، جملة التخصيص المنطقية لها الشكل التالى :

تعبــير منطق = متغير منطق

لذلك فإن :

```
LOGICAL B, C
B = X.LT.Y
C = B
```

تخصص القيمة .TRUE. أو .FALSE. إلى B تبعاً لكون $X < Y$ تحققت أو لم تتحقق وبعد ذلك تخصص قيمة B هذه إلى C . مثلا :

```
LOGICAL D
D = (R.GT.S).OR.(X.LE.Y)
```

تخصص القيمة .TRUE. إلى D إذا تحققت $R > S$ أو $X \leq Y$ (أو تحقق كلاهما) وتخصص القيمة .FALSE. إلى D فيما عدا ذلك . (لاحظ أنه من الممكن أن تستخدم الأقواس بالطريقة التقليدية) .

يمكن أيضاً أن تستخدم جملة DATA لتخصيص قيمة منطقية . فمثلا :

```
LOGICAL A, B
DATA A/.TRUE./, B/.FALSE./
```

تخصص القيمة .TRUE. إلى A و .FALSE. إلى B أثناء الترجمة (نناقش جملة DATA بالتفصيل فى قسم ١٠ – ٢) .

يمكن أن تكون القيمة المزاملة لأسم البرنامج الفرعى FUNCTION منطقية . وتكون نتيجة البرنامج الفرعى FUNCTION القيمة .TRUE. فقط إذا كانت X عنصراً فى المجموعة المتراصة A .

```
      LOGICAL FUNCTION CHECK(A, N, X)
      DIMENSION A(N)
      CHECK = .TRUE.
      DO 99 I = 1, N
           IF(X.EQ.A(I)) RETURN
99    CONTINUE
      CHECK = .FALSE.
      RETURN
      END
```

لاحظ إعلان النوع فى جملة تعريف FUNCTION الذى يخبر الحاسب أن CHECK متغير منطقى وليس متغيراً حقيقياً .

٩ – ٩ حقل L

يمكن أن تخزن أو تطبع المتغيرات المنطقية بجمل READ و WRITE . ويتم هذا عن طريق حقل L الذى صيغته لها المواصفات المبينة فى الشكل التالى :

Lw

حيث تشير *w* إلى عرض الحقل . (يمكن أن نستخدم أيضاً *rLw* ، حيث *r* هى معامل التكرار) ويستخدم حقل L كما يلى :

(أ) إدخـــال

إفرض أن صيغة الكود *Lw* تناظر المتغير المنطقى A . وعلى ذلك تخصص إلى A القيمة .TRUE. إذا كان أول حرف غير خال فى الحقل المشار إليه هو حرف T . إما إذا كان أول حرف خال هو F أو أن الحقل بأكله يتكون من مسافات فتخصص .FALSE. إلى A . (تعتبر بعض مترجمات الفورتران أى رمز أمامى غير T أو F هو .FALSE . فى حين ستعطى مترجمات أخرى رسالة خطأ) .

(ب) الإخــــراج

فى الخرج سيطبع الحرف T أو F مضبطة من اليمين فى الحقل الخاص بها تبعاً للقيمة التى يأخذها المتغير المنطقى المناظر .TRUE. أو .FALSE. كمثال أفرض A و B متغيرين منطقيين يحتويان .TRUE . و .FALSE . على الترتيب . وعلى ذلك فأمر الخرج التالى :

```
      WRITE(6, 10) A, B
10    FORMAT(1X, L5, 2X, L3)
```

سيطبع T فى عمود 5 و F فى عمود 10 . بذلك فإن كود الصيغة *Lw* سيطبع عدد 1 — *w* من الفراغات قبل طباعة الحرف T أو F .

مثال ٩ — ٤

(أ) إفرض بطاقة بيانات مثقبة كالتالى :

```
          1                   2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
THE  FIRST  DAY
```

بالإضافة إلى ذلك ، إفرض أمر الإدخال هو :

```
      LOGICAL A, B, C
      READ(5, 10) A, B, C
   10 FORMAT(L3, 2L5)
```

عروض الحقول 3 و 5 و 5 على الترتيب وأول حرف غير خال فى الأعمدة من 1 إلى 3 هو T وفى الأعمدة 4 إلى 8 هو F وفى الأعمدة من 9 إلى 13 هو T . من ثم . تخصص .TRUE. ، إلى المتغيرات المنطقية A و C وتخصص .FALSE. إلى B .

(ب) إفرض جملة FORMAT فى جزء (أ) تم تغييرها إلى :

```
   10 FORMAT(3L5)
```

هنا T و I و D هى أول حروف غير خالية فى الحقول الثلاثة . من ثم ستخصص بعض المترجمات .TRUE. إلى A و .FALSE. إلى C, B ولكن ستعطى مترجمات أخرى رسالة خطأ .

(ـ) أوجد الخرج لجزء التالى من البرنامج :

```
      LOGICAL X, Y, Z
      J = 5
      X = .FALSE.
      Y = 3*J.GT.12
      Z = Y.AND.X
      WRITE(6, 20) X, Y, Z
   20 FORMAT(1X, 3L10)
```

تخصص .FALSE. إلى X ، بواسطة السطر الثالث . حيث تتحقق 12 < 15 — = 3*J وتخصص . TRUE . إلى Y بواسطة السطر الرابع . ورغم أن Y تتحقق و X لاتتحقق فإنه يتم تخصيص.FALSE . إلى Z بواسطة السطر الخامس . وعلى ذلك تطبع F و T و F فى الأعمدة 10 و 20 و 30 على الترتيب .

٩ — ١٠ التدرج الهرمى للعمليات الحسابية

حيث أن الأدوات المنطقية تشمل تعبيرات مترابطة والتى بدورها تشمل تعبيرات حسابية ومعاملات حسابية ، فمن المهم أن نفهم التدرج الهرمى الكامل للعمليات . وفيما يلى الترتيب الذى تجرى به العمليات :

١ — العمليات الحسابية

(أ) الأس (* *) .

(ب) ضرب وقسمة (* و/) .

(ـ) جمع وطرح (+ و −) .

٢ — معاملات الترابط .

٣ — معاملات منطقية :

(أ) .NOT.
(ب) .AND.
(ح) .OR.

(وكما ذكرنا فى قسم ٢ — ٦ ، سنعامل الجمع والطرح الأحادى على نفس مستوى الجمع والطرح الثنائى) .

نلاحظ أن العمليات الحسابية تنفذ أولا ، ثم تجرى العمليات الترابطة قبل العمليات المنطقية . أما بين العمليات الست الترابطة ، فليس هناك أولويات . وبالتحديد تنفذ معاملات الترابط بالترتيب التى تظهر به من اليسار إلى اليمين . وكما فى التعبيرات الحسابية ، فإنه من الممكن أن نغير هذا التدرج الهرمى باستخدام الأقواس . وعلاوة على ذلك ، فإننا ننصح بشدة باستخدام الأقواس كلما كان هذا الاستعمال يسهل فهم ترتيب العمليات . فمثلا ننصح بكتابة :

(M.GE.1).OR.(X.NE.Y)　　　(K.GT.10).OR.((X.GE.0.2).AND.(Y.EQ.3.0))

بدلا من

M.GE.1.OR.X.NE.Y　　و　　K.GT.10.OR.X.GE.0.2.AND.Y.EQ.3.0

رغم أنهما متكافئتان .

مسائل محلولة

بيانات حرفية

٩ — ١ باعتبار السعة الحرفية هى M=4 أوجد البيانات المخصصة لكل من I و J بعد تنفيذ الجملة

DATA I/'NO'/, J/'THE END'/

تخزن الـ NO مضبطة من اليسار فى I مع عدد 2 = 2 — 4 من المسافات مضافة على اليمين ، أما فى J فيخزن فقط عدد 4 حروف من (M = 4) أقصى اليسار (بما فيها المسافات) من السلسلة THE END :

I | N | O | b | b |　　J | T | H | E | b |

٩ — ٢ إفرض أن M = 4 وإن بطاقة بيانات مثقبة كالتالى :

```
          1                   2
1234567890123456789012345
JOHN PAUL JONES
```

أوجد البيانات فى المخزن إذا كان أمر الإدخال كالتالى :

(أ)
```
   READ(5, 10) I
10 FORMAT(A3)
```

(ب)
```
   READ(5, 20) L
20 FORMAT(A7)
```

(جـ)
```
   DIMENSION K(4)
   READ(5, 30) K
30 FORMAT(4A4)
```

(د)
```
   DIMENSION J(3)
   READ(5, 40) J
40 FORMAT(3A5)
```

(أ) حيث أن عرض الحقل هو ٣ فتخصص لـ I سلسلة الحروف JOH الموجودة فى الأعمدة من 1 إلى 3 . وسوف تخزن فى I مضبطة من اليسار مع إضافة مسافة واحدة على اليمين ، كما فى الشكل ٩ – ١ (أ) .

(ب) حيث أن عرض الحقل هو 7 فتخصص سلسلة الحروف JOHNPA الموجودة فى الأعمدة من 1 إلى 7 إلى L . وسوف تبتر من اليسار وسيخزن فى L الحروف الأربعة الموجودة أقصى اليمين فقط ، كما فى الشكل ٩ – ١ (ب) .

(جـ) حيث أن عرض كل حقل هو 4 فتخزن البيانات كما فى الشكل ٩ – ١ (جـ) .

(د) حيث أن عرض كل حقل هو 5 .فتخصص السلاسل 'JOHN$_b$' و 'PAUL$_b$' و 'JONES' إلى عناصر J . ومع ذلك حيث أنه يوضع فى التخزين الحروف الأربعة فقط الموجودة أقصى اليمين ، فتخزن البيانات كما فى الشكل ٩–١ (د) .

(أ)

I	J	O	H	b

(ب)

L	N	b	P	A

(جـ)

K(1)	J	O	H	N
K(2)	b	P	A	U
K(3)	L	b	J	O
K(4)	N	E	S	b

(د)

J(1)	O	H	N	b
J(2)	A	U	L	b
J(3)	O	N	E	S

شكل ٩ – ١

٩ – ٣ إفرض أن M = 4 وتم تثقيب بطاقة كالآتى :

```
                  1                   2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
E R I C S O N   M O R T Y
```

(أ) أوجد البيانات فى المخزن إذا كان أمر الإدخال هو :

```
   INTEGER A, B, C, D
   READ(5, 10) A, B, C, D
10 FORMAT(4A4)
```

(ب) أوجد المخرج إذا كان أمر الإخراج هو :

(١)
```
   WRITE(6, 20) A, B
20 FORMAT(1X, A8, A8)
```

(٢)
```
   WRITE(6, 30) C, D, B, A
30 FORMAT(1X, 2A1, 1X, 2A4)
```

(أ) تخزن البيانات الحرفية ، كالآتى :

A	E	R	I	C
B	S	O	N	b
C	M	O	R	T
D	Y	b	b	b

(ب) (i) يحذف إشارة التحكم فى العرب 1X ، تكون عروض الحقول لـ A و B هما 8 و 8 . وتطبع المحتويات من المخزن مضبطة من جهة اليمين فى الحقول الخاصة بها كما يلى :

$_{bbbb}$ERIC$_{bbbb}$SON$_b$

لاحظ أن $w - M = 8 - 4 = 4$ وهى عدد المسافات التى تسبق ERIC و SON$_b$.

(ii) عروض الحقول A و B و C و D هى 4 و 4 و 1 و 1 على الترتيب . تبتر السلاسل الحرفية فى C و D من اليمين ، ومن ثم ، يطبع :

MY$_b$SON$_b$ERIC

٩ — ٤ إفرض أن M = 4 وأن بطاقتى البيانات الأولين ثقبتا كالتالى :

```
                1                   2                   3                   4
      1234567890123456789012345678901234567890123456789012345
البطاقة الأولى        JEFFERSON           THOMAS
البطاقة الثانية       WASHINGTON          GEORGE
```

أوجد الخرج لكل من أجزاء البرامج التالية :

(أ)

```
      DIMENSION NAME(30)
      DO 100 I = 1, 2
         READ(5, 60) NAME
60       FORMAT(5X, 30A1)
         WRITE(6, 60) NAME
100   CONTINUE
```

(ب)

```
      DIMENSION NAME(10)
      DO 100 I = 1, 2
         READ(5, 35) NAME
35       FORMAT(5X, 10A4)
         WRITE(6, 45)
45       FORMAT(5X, 10A3)
100   CONTINUE
```

(جـ)

```
      DO 100 I = 1, 2
         READ(5, 25)
25       FORMAT(5X, 30HAAAAAAAAAABBBBBBBBBBCCCCCCCCCC)
         WRITE(6, 25)
100   CONTINUE
```

(أ) يخزن كل حرف فى عنصر من المجموعة المتراصة NAME ثم يطبع الحرف بعد ذلك . من ثم ، يظهر الخرج كما فى بطاقة البيانات :

```
JEFFERSON      THOMAS
WASHINGTON     GEORGE
```

(ب) لاحظ أنه تم تخصيص أربعة حروف لكل عنصر من المجموعة المتراصة NAME ولكن ثلاثة حروف فقط ، أى الثلاثة الأولى ، الحروف هى التى تطبع . ومن ثم سيظهر الخرج كما يلى :

```
JEFERSN      HOMS
WASINGON     EORE
```

(جـ) تحل الـ 30 حرف فى الأعمدة 6 — 35 محل الحروف التالية لـ 30 H فى جملة FORMAT وبعد ذلك تطبع الحروف . من ثم يظهر الخرج كما هو فى بطاقة البيانات :

```
JEFFERSON      THOMAS
WASHINGTON     GEORGE
```

٩ — ٥ إفرض أن جملة كاملة S قد ثقبت على بطاقة واحدة إبتداء من عمود 1 . احسب عدد الكلمات N فى S .

بفرض أن الكلمات تنفصل عن بعضها بواسطة مسافة واحدة وأن الجملة تنتهى بنقطة ، فإن N تساوى عدد المسافات التى تسبق النقطة مضافاً إليها واحد . وحيث أن البطاقة تحتوى 80 عموداً ، فنخزن جملتنا فى مجموعة متراصة خطية STRING بها 80 عنصر . وفيما يلى البرنامج :

```
      INTEGER STRING(80), BLANK, PERIOD
      DATA BLANK/' '/, PERIOD/'.'/
      READ(5, 10) STRING
   10 FORMAT(80A1)
      N = 0
      DO 100 K = 1, 80
C           TEST IF IT IS A BLANK CHARACTER
            IF(STRING(K).EQ.BLANK) N = N + 1
C           TEST IF IT IS THE END OF THE SENTENCE
            IF(STRING(K).EQ.PERIOD) GO TO 55
  100 CONTINUE
   55 N = N + 1
      WRITE(6, 20) N
   20 FORMAT(1X, 'THE NUMBER OF WORDS = ', I3)
      STOP
      END
```

٩ — ٦ تم تثقيب قائمة لاتتجاوز 500 اسم على بطاقات ، كل إسم على بطاقة ، تم تثقيب الأسماء الأخيرة فى الأعمدة من 1 إلى 15 وتم تثقيب الأسماء الأولى فى الأعمدة من 17 إلى 32 وأضيفت بطاقة خلفية بها XXX مثقبة فى الأعمدة من 1 إلى 3 . بفرض أن M = 4 اكتب جزء برنامج فورتران ليحسب عدد الأسماء N ويخزنها فى مجموعة متراصة .

حيث أن السعة M = 4 فيحتاج كل إسم إلى العدد 8 = 32/4 من أماكن التخزين على الأقل وذلك لتخزين الحروف 32 الممكنة ومن ثم ستخزن الأسماء فى NAME مجموعة متراصة 8 × 500 اسم واحد فى كل صف من المجموعة المتراصة . وعلاوة على ذلك ، كل مرة نقرأ بطاقة ، نتحقق من ظهور XXX فى أول مكان تخزين . وفيما يلى البرنامج :

```
      DIMENSION NAME(500, 8)
      DATA JJ/'XXX'/
      N = 0
      DO 100 K = 1, 500
            READ(5, 10) (NAME(K, L), L = 1, 8)
10          FORMAT(8A4)
            IF(NAME(K, 1).EQ.JJ) GO TO 88
            N = N + 1
100   CONTINUE
88    ....................
```

٩ — ٧ إفرض أن M=4 و NAME مجموعة متراصة (IXJ) تحتوى إسماً واحداً فى كل صف اكتب برنامج فرعى SUBROUTINE :

```
ALPHA(NAME, I, J, K, L)
```

يبدل الأسماء رقم K و L إن لم يكونا فى الترتيب الأبجدى .

تفرز الأسماء أبجدياً كما لو كنا نفرز مكانهم الرقمى ، فيما عدا الإسمين الأولين إذا كان لهما نفس الحروف الأربعة الأولى ، أى ،

```
NAME(K, 1) = NAME(L, 1)
```

إذن يجب أن نقارن الحروف الأربعة الثانية ، وهكذا . وفيما يلى البرنامج :

```
      SUBROUTINE ALPHA(NAME, I, J, K, L)
      DIMENSION NAME(I, J)
      DO 99 N = 1, J
            IF(NAME(K, N).GT.NAME(L, N)) GO TO 50
            IF(NAME(K, N).LT.NAME(L, N)) GO TO 40
99    CONTINUE
40    RETURN.
50    DO 88 M = N, J
            JSAVE = NAME(K, M)
            NAME(K, M) = NAME(L, M)
            NAME(L, M) = JSAVE
83    CONTINUE
      RETURN
      END
```

لاحظ أن تأثير الجملة المرقمة 50 يكافىء :

```
50    DO 88 M = 1, J
```

وسوف ندع القارىء يعطى السبب لذلك .

٩ — ٨ إفرض أن عدد N من الأسماء مخزن فى الذاكرة فى مجموعة متراصة NAME (500 × 8) إسم واحد فى كل صف . أكتب جزء برنامج لفرز المجموعة المتراصة NAME أبجدياً .

إفرض A_1 و A_2 و ... و A_N هى الأسماء . سنستخدم النظام الحسابى (الخوارزم) للفرز الفقاعى الذى تمت مناقشته فى مسألة ٦ — ٢ وقسم ٨ — ٢ ، فيما عدا أننا هنا نرتب الأسماء أبجدياً فضلا عن ترتيب الأرقام . باستخدام البرنامج

الفرعى ALPHA فى مسألة ٩ – ٧ ، نفرز A_1 و A_2 أبجدياً ثم A_2 و A_3 وهكذا إلى A_{N-1} و A_N . بعد هذا المرور خلال العناصر ، يصبح الإسم الصحيح فى المركز الأخير . تكرر هذه العملية مع الأسماء A_1 و A_2 و . . . و A_{N-1} بعد المرور خلال الأسماء ، يصبح الإسم الصحيح فى المركز قبل الأخير ونستمر فى ذلك وبعد عدد N — 1 من اللفات ، ستكون الأسماء فى الترتيب الأبجدى .

لترجمة النظام الحسابى (الخوارزم) إلى الفورتران ، نستخدم حلقات DO المتداخلة ، يتغير الدليل الخارجى I من 1 إلى N — 1 والدليل الداخلى J من 1 إلى N — I وفيما يلى البرنامج :

```
    NN = N - 1
    DO 99 I = 1, NN
          JJ = N - I
          DO 88 J = 1, JJ
                CALL ALPHA(NAME, 500, 8, J, J + 1)
88        CONTINUE
99  CONTINUE
```

تغيرات منطقية وموصلات

٩ – ٩ إفرض أن بطاقة بيانات ثقبت كما يلى :

```
                  1                   2                   3
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 N O W   I S   T H E   T I M E   T O   R E T U R N
```

بفرض أن أى حرف غير خال فى الحقل غير T سيقبل على أنه . FALSE . (أنظر ص ٢٦٤) ، أوجد قيم I و J و K و X و Y إذا كان الأمر هو :

```
    LOGICAL I, J, K, X, Y
    READ(5, 10) I, J, K, X, Y
10  FORMAT(L6, L3, 3L6)
```

عروض الحقول هى 6 و 3 و 6 و 6 و 6 على الترتيب . أول حرف غير خال فى الأعمدة من 1 إلى 6 هو N وفى الأعمدة من 7 إلى 9 هو T فى الأعمدة من 10 إلى 15 هو E فى الأعمدة من 16 إلى 20 هو T وفى الأعمدة من 22 إلى 27 هو T . من ثم ، نقرأ . FALSE . فى I و K وتقرأ . TRUE . فى J و X و Y .

٩ – ١٠ اكتب مايلى بالفورتران :

(أ) توقف إذا كانت $J = K$ و $N > 15$.

(ب) إذهب إلى الجملة المرقمة 71 . إذا كانت $X \geq Y$ أو $A + B < C$.

(حـ) إذهب إلى الجملة المرقمة 72 إذا كانت A ليست أكبر من B .

(د) توقف أن لم تكن $A < B$ و $C \geq D$.

تذكر أن "*and*" و "*or*" و "*not*" يرمز إليها فى الفورتران بـ «.AND.» «.OR.» ، و «.NOT.» . على الترتيب.

(أ) IF(J.EQ.K.AND.N.GT.15) STOP
(ب) IF(X.GE.Y**2.OR.(A + B).LT.C) GO TO 71
(جـ) IF(.NOT.A.GT.B) GO TO 72
(د) IF(.NOT.(A.LT.B.AND.C.GE.D)) STOP

٩ – ١١ إفرض أن K تحتوى على 5 وأن A و B متغيرات منطقية . أوجد القيمة النهائية لـ A بعد كل جزء برنامج فورتران مما يلى :

(أ)
```
A = .FALSE.
B = 2*K.LT.15
A = A.OR.B
```

(ب)
```
B = .TRUE.
A = 5.NE.K
A = A.AND.B
```

(جـ)
```
B = 10.LT.3*K
A = K**2.GT.4*K
IF(B) A = .NOT.A
```

(أ) لاتتحقق A بواسطة السطر الأول وتتحقق B بواسطة السطر الثانى ، من ثم ، تتحقق قيمة A.OR.B وتصبح هى القيمة النهائية لـ A .

(ب) تتحقق B بواسطة السطر الأول ، ولا تتحقق A بواسطة السطر الثانى ، من ثم ، لاتتحقق قيمة A.AND.B وتصبح هى القيمة النهائية لـ A .

(جـ) تتحقق B بواسطة السطر الأول ، وتتحقق A بواسطة السطر الثانى وحيث أن B تحققت وأن A = .NOT.A فمن ثم تصبح قيمة A لاتتحقق .

٩ – ١٢ إفرض أن A و B و C متغيرات منطقية تحتوى .TRUE. و .FALSE. و .TRUE. على الترتيب . أوجد الخرج لكل أمر مما يلى :

(أ)
```
   WRITE(6, 10) A, B, C
10 FORMAT(1X, 3L8)
```

(ب)
```
   WRITE(6, 20), A, B, C
20 FORMAT(1X, L4, 3X, L4, 3X, L4)
```

فى الخرج يطبع كود الشكل *Lw* عدد $w - 1$ من المسافات متبوعة بـ T أو F تبعاً لما إذا كان المتغير المنطقى المناظر يتحقق أولا يتحقق من ثم :

(أ) تطبع T و F و T فى الأعمدة 8 و 16 و 24 على الترتيب .

(ب) تطبع T و F و T فى الأعمدة 4 و 11 و 18 على الترتيب .

مسـائل تكميلية

بيـانات حرفيـة

٩ – ١٣ بفرض أن سعة الحرف M = 4 أوجد البيانات فى المخزن بعد تنفيذ ما يلى :

```
DATA I/'YES'/, J/'GO NOW'/
```

٩ – ١٤ بفرض أن M = 4 و تم تثقيب بطاقة بيانات كالتالى :

```
                  1                   2
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
  L E N A     A U D R E Y     E R I C
```

اوجد البيانات فى المخزن إذا كان أمر الإدخال هو :

(أ)
```
    READ(5, 10) J
10  FORMAT(A2)
```

(ب)
```
    READ(5, 20) J
20  FORMAT(A7)
```

(جـ)
```
    DIMENSION K(5)
    READ(5, 30) K
30  FORMAT(5A4)
```

(د)
```
    DIMENSION L(4)
    READ(5, 40) L
40  FORMAT(4A5)
```

٩ — ١٥ إفرض أن M = 4 وتم تخزين LONDON ، فى IA و IB كالتالى :

IA | L | O | N | D | IB | O | N | b | b |

أوجد الخرج إذا كانت جملة WRITE هى :

```
WRITE(6, 55) IA, IB
```

وجملة FORMAT المصاحبة هى :

(أ) `55 FORMAT(1X, 2A4)`

(ب) `55 FORMAT(1X, 2A7)`

(جـ) `55 FORMAT(1X, A2, A3)`

(د) `55 FORMAT(1X, A3, A2)`

٩ — ١٦ إفرض أن M = 4 وتم تثقيب أول بطاقتى بيانات كالتالى :

```
                    1                   2                   3
          1234567890123456789012345678901234 5
بطاقة 1             BROWN          ROBERT
بطاقة 2             JOHNSON        WILLIAM
```

أوجد الخرج لكل جزء برنامج مما يلى :

(أ)
```
    DIMENSION NAME(6)
    DO 99 K = 1, 2
          READ(5, 15) NAME
15        FORMAT(10X, 6A4)
          WRITE(6, 15) NAME
99  CONTINUE
```

(ب)
```
    DIMENSION NLAST(3), NFIRST(3)
    DO 99 K = 1, 2
          READ(5, 15) NLAST, NFIRST
15        FORMAT(10X, 6A4)
          WRITE(6, 15) NLAST, NFIRST
99  CONTINUE
```

(ﺟ)

```
    DO 99 K = 1, 2
          READ(5, 15)
15        FORMAT(10X, 24HAAAAAAAABBBBBBBBCCCCCCCC)
          WRITE(6, 15)
99  CONTINUE
```

(د)

```
    DIMENSION NAME(6)
    DO 99 K = 1, 2
          READ(5, 15) NAME
15        FORMAT(10X, 6A4)
          WRITE(5, 25) NAME
25        FORMAT(10X, 6A2)
99  CONTINUE
```

٩ — ١٧ نفذ مسألة ٩ — ٥ ، عدا أن كلمات الجملة الآن مفصولة بمسافة أو أكثر .

٩ — ١٨ إفرض أنك أعطيت مجموعة بطاقات لاتتجاوز 80 بطاقة . تحتوى كل بطاقة على اسم طالب فى الأعمدة من 1 إلى 32 وحروف الدرجة فى عمود 34 . حروف الدرجات المقبولة هى A و B و C وD و F و I (لمن لم يستكمل المنهج) إفرض أن بطاقة خلفية قد أضيفت ومثقب عليها XXX فى الأعمدة من 1 إلى 3 اكتب برنامج :

(أ) يحسب عدد الطلبة الحاصلين على كل درجة .

(ب) يطبع أسماء الطلبة فى كل مستوى درجة .

٩ — ١٩ إفرض أن قصيدة P تم تثقيبها فى مجموعة من البطاقات حيث يثقب على كل بطاقة سطر من القصيدة ، وافرض أن المجموعة لها بطاقة خلفية مثقب عليها XXX فى الأعمدة من 1 إلى 3 . (افرض أن القصيدة P ليس بها أكثر من 20 سطراً) اكتب البرنامج الذى يطبع القصيدة P وعدد البطاقات فى المجموعة ، أى ، عدد الأسطر فى P .

٩ — ٢٠ للقصيدة P فى المسألة ٩ — ١٩ ، أكتب البرنامج الذى يحسب عدد المرات التى ظهرت فيها الحروف A و E و I و O و U فى القصيدة P .

٩ — ٢١ فى مسألة ٩ — ١٩ ، اكتب البرنامج الذى يحسب عدد الأسطر التى تبدأ بكلمة THE فى القصيدة P (إفرض أن كل سطر فى القصيدة يبدأ فى عمود 1 من بطاقة البيانات) .

٩ — ٢٢ فى مسألة ٩ — ١٩ ، أكتب البرنامج الذى يحسب عدد المرات التى تظهر فيها كلمة THE فى القصيدة P (افرض أن كلمة THE تكون متبوعة دائماً بمسافة خالية ومسبوقة دائماً بمسافة خالية فيما عدا عند بداية السطر) .

٩ — ٢٣ إفرض أنه تم تثقيب قصة قصيرة SS على مجموعة بطاقات ، وافرض أن المجموعة لها بطاقة خلفية مثقب عليها XXX فى الأعمدة من 1 إلى 3 . اكتب البرنامج الذى يطبع القصة القصيرة SS وعدد الأسطر فى SS . (افرض أن القصة ليس بها أكثر من 400 سطراً باستبعاد العنوان المثقب فى أول بطاقة واسم الكاتب المثقب فى ثانى بطاقة) .

٩ — ٢٤ فى مسألة ٩ — ٢٣ ، اكتب البرنامج الذى يحسب عدد الفقرات فى القصة القصيرة SS (أفرض أن كل فقرة مثقبة على بطاقة جديدة من عمود 6 وغير ذلك كل البطاقات مثقبة إبتداء من عمود 1) .

٩ — ٢٥ إفرض عنوان به أقل من 80 حرفاً مثقب على بطاقة إبتداء من عمود 1 . اكتب البرنامج الذى يطبع العنوان بحيث يكون متوسطاً فى الأعمدة من 1 إلى 80 .

٩ – ٢٦ إفرض قائمة N من الأسماء $(N \leq 500)$ خزنت فى مجموعة مترا صة (500×8) باسم NAME إسم واحد لكل صف . إفرض مجموعة متراصة خطية صحيحة PERSON (1) و ... ... و PERSON(8) تحتوى على إسم آخر اكتب جزء البرنامج الذى يحدد ما إذا كان PERSON موجوداً فى NAME . إذا كانت الإجابة نعم ، أوجد المكان L للاسم ، وضع $L = 0$ فيما عدا ذلك .

٩ – ٢٧ إفرض قائمة N من الأسماء $(N < 500)$ خزنت فى مجموعة متراصة (500×8) باسم NAME والمراد إضافة إسم مخزن فى المجموعة المتراصة الصحيحة PERSON(1) و ... و PERSON(8) فى الصف L من المجموعة المتراصة NAME وذلك بتحريك الأسماء فى الصفوف L و $L + 1$ و ... و N إلى أسفل مكان واحد . أكتب جزء برنامج لإنجاز هذه المهمة .

٩ – ٢٨ إفرض قائمة N من الأسماء $(N \leq 500)$ خزنت فى مجموعة متراصة (500×8) باسم NAME مرتبة ترتيباً أبجدياً . استخدم طريقة البحث الثنائى لتحدد ما إذا كان الاسم المخزن فى المجموعة المتراصة الصحيحة PERSON(1) و ... و PERSON(8) موجوداً فى المجموعة المتراصة NAME . إذا كان كذلك ، اجعل المتغير المنطق IN به القيمة TRUE . وأوجد مكان الصف L . إن لم يكن كذلك ، اجعل IN به القيمة .FALSE. وأوجد مكان الصف L حيث يجب أن يضاف هذا الإسم ، ثم أضف الإسم كما فى المسألة ٩ – ٢٧ .

متغيرات منطقية وروابط

٩ – ٢٩ إفرض المتغيرات المنطقية الوحيدة فى البرنامج هى A و B و C و D اكتشف الأخطاء ، إن وجدت ، فى كل جملة :

(أ) A = FALSE (جـ) C = R.GT.(A + 2.0) (هـ) F = C.AND.D
(ب) B = Y.GT.X (د) D = A.OR.B (و) A = A.AND..TRUE.

٩ – ٣٠ إفرض أنه تم تثقيب بطاقة بيانات كما يلى :

```
         1         2
1234567890123456789012345
TOM LEFT. IN THE NIGHT
```

بفرض أن المترجم يعتبر أول حرف غير T لايتحقق . أوجد الخرج لكل جزء برنامج مما يلى :

(أ)
```
      LOGICAL A(6)
      READ(5, 15) A
   15 FORMAT(5L4)
      WRITE(6, 25) A
   25 FORMAT(1X, 5L4)
```

(ب)
```
      LOGICAL X, Y, Z
      READ(5, 15) X, Y, Z
   15 FORMAT(3X, 3L4)
      WRITE(6, 25) X, Y, Z
   25 FORMAT(1X, 3L4)
```

٩ – ٣١ اكتب الآتى بالفورتران .

(أ) توقف إذا كانت $A > B$ و $C \leq 10$.

(ب) إذهب إلى الجملة رقم 100 إذا كانت A موجبة أو $J \not\leq L$.

(جـ) إذهب إلى الجملة رقم 200 إذا كانت $7 < X < 10$.

(د) توقف أن لم يتحقق الشرط $A \leq B$ و $X \geq Y$.

٩ – ٣٢ إفرض أن A وB وC متغيرات منطقية ولها القيم .FALSE. و .FALSE. و .TRUE. على الترتيب . أوجد قيمة كل تعبير منطقي مما يأتي :

(أ) A.OR.C (د) NOT.A.AND.B (ز) A.OR.B.AND.C
(ب) B.AND.C (هـ) NOT.(A.AND.B) (ح) .NOT.(.NOT.B.OR.A)
(جـ) A.OR..NOT.C (و) A.AND.B.OR..NOT.C (ط) A.OR.C.AND.B.OR.C

٩ – ٣٣ إفرض أن J تحتوى على 10 وأن A وB متغيرات منطقية . أوجد القيمة النهائية لـ A بعد كل جزء برنامج مما يلي :

(أ)
```
A    .TRUE.
B = J.LT.5
A = A.AND.B
```

(ب)
```
B = .TRUE.
A = .FALSE.
IF(.NOT.A) B = .FALSE.
A = A.OR.B
```

(جـ)
```
A = J + 10.EQ.2*J
B = .TRUE.
IF(A.OR..NOT.B) J = J + 10
A = J.GT.15
```

(د)
```
A = .TRUE.
B = .NOT.A
A = B.OR.J**2.EQ.100
```

٩ – ٣٤ إفرض أن كل من J وK وL تحتوى على 10 و 20 و30 على الترتيب . أوجد قيمة كل تعبير منطقي .

(أ) 2*J.EQ.K.AND.K.LE.L
(ب) NOT.J.GT.K.OR.L.LT.5
(جـ) K.EQ.10.AND..NOT.J.LT.L – 15
(د) NOT.(5.EQ.J – 5.AND.2*K.EQ.J + L)

٩ – ٣٥ اكتب تعبيراً منطقياً بالفورتران لكل شرط :

(أ) A أقل من كل من X وY وZ .

(ب) تقل A عن X وY وZ على الأقل بواحد .

(جـ) $X > 2$ ولكن $Y \leq 3$.

(د) كل من A و B موجب ، أو أحدهما أكبر من 100 .

(هـ) $0 < X < 1$ و $Y \geq Z$.

اجابات للمسائل التكميلية المختارة

٩ – ١٣

YES$_b$ في I و GO$_b$N في J

٩ – ١٤ (أ) LE$_{bb}$ في J ، (ب) A$_{bb}$A في J (جـ) LENA في K(1) ، $_{bb}$AU في K(2) ، DERY في K(3) ، $_{bb}$ER في K(4) و IC$_{bb}$ في K(5) . (د) ENA$_b$ في L(1) ، AUDR في L(2) ، Y$_{bb}$E في L(3) و IC$_{bb}$ في L(4)

٩ – ١٥

(أ) LONDON$_{bb}$ (ب) $_{bbb}$LOND$_{bbb}$ON$_{bb}$ (جـ) LOON$_b$ (د) LONON

٩ – ١٦ (أ) ، (ب) ، (جـ) يظهر الخرج كما في بطاقات البيانات .

(د) BRN$_{bbb}$RORT$_{bb}$ و JOSO$_{bb}$WHA$_{bb}$.

٩ – ٢٩ (أ) A = .FALSE. (ب) لاتوجد أخطاء . (جـ) A + 2.0 غير مقبولة حيث أن A متغيراً منطقياً .
(هـ) غير مقبولة حيث أن F ليست متغيراً منطقياً (و) لاتوجد أخطاء .

٩ – ٣٠

(أ) T, F, T, F, F, T (ب) T, T, F

٩ – ٣١
(أ) IF(A.GT.B.AND.C.LE.10.0) STOP
(ب) IF(J.NE.L.OR.A.GT.0.0) GO TO 100
(جـ) IF(7.0.LT.X.AND.X.LT.10.0) GO TO 200
(د) IF(.NOT.(A.LE.B.AND.X.GE.Y)) STOP

٩ – ٣٢

(أ) T (ب) F (جـ) F (د) F (هـ) T (و) F (ز) F (ح) F (ط) T

٩ – ٣٣

(أ) F (ب) F (جـ) T (د) T

٩ – ٣٤

(أ) T (ب) T (جـ) F (د) F

٩ – ٣٥
(أ) A.LT.X.AND.A.LT.Y.AND.A.LT.Z
(ب) A.LT.X.OR.A.LT.Y.OR.A.LT.Z
(جـ) X.GT.2.0.AND.Y.LE.3.0
(د) A.GT.0.0.AND.B.GT.0.0.OR.(A.GT.100.0.OR.B.GT.100.0)
(هـ) (0.0.LT.X.AND.X.LT.1.0).AND.(Y.GE.Z)

الفصل العاشر

ملامح اضافية للادخال / الاخراج

١٠ — ١ مقدمـــــة

هناك ملامح أخرى للإدخال /ا لإخراج لم تناقش بعد . ففى الفصل الثالث ، ناقشنا مواصفات حقل —I وحقل —F وحقل —E وحقل —X وفى الفصل التاسع ناقشنا مواصفات حقل —A وحقل —H وحقل —L وسنقدم فى هذا الفصل عدة ملامح جديدة ، بالإضافة إلى أننا سوف نناقش كيف تخرج (تطبع) الرسم البيانى لدالة .

أولا ، نريد أن نذكر القارىء أن أول حرف فى أى (سجل) خرج لايطبع بواسطة آلة الطباعة ولكنه يذكر فقط التحكم فى العربة . ومثل هذه المعلومة ومعلومات أخرى عن الإدخال / الإخراج يمكن أن نجدها فى الفصلين ، الثالث والتاسع .

١٠ — ٢ جملة DATA

لتخزين الثابت π ، مثلا يمكن أن نستخدم جملة التخصيص :

PI = 3.14159

حيث أن هذه الجملة منفذة ، سيخصص الثابت 3.14159 إلى PI كلما تم تنفيذ الجملة . وحيث أن π لاتتغير (ثابتة) طوال البرنامج ، فنود أن نضع القيمة 3.14159 فى PI مرة واحدة فقط ويمكن إنجاز ذلك بواسطة جملة DATA كما يلى :

DATA PI/3.14159/

أى أن كلمة DATA يتبعها إسم المتغير ، وبعد ذلك تحاط قيمته بشرطتين مائلتين وحيث أن جملة DATA جملة غير منفذة ، لذا يجب أن توضع قبل أى جملة منفذة فى البرنامج .

يمكن أن نستخدم جمل DATA لإعطاء قيم إبتدائية لمتغيرات معينة ، مثلا يمكن أن نعطى قيمة إبتدائية للعداد KOUNT ، ولتكن واحدا وكذلك المجموع التراكمى SUM ولتكن صفراً . يمكن أن نجرى ذلك كما يلى :

```
DATA KOUNT/1/
DATA SUM/0.0/
```

يمكن أن نستخدم جملة DATA واحدة لتحديد عدة قيم ، فمثلا :

```
DATA KOUNT/1/, SUM/0.0/
```

أو كبديل لذلك :

```
DATA KOUNT, SUM/1, 0.0/
```

الجملة الأخيرة لها الشكل العام التالى :

DATA قائمة من المتغيرات / قائمة من الثوابت /

تفصل هذه المتغيرات بواسطة فصلات ، وتفصل الثوابت بواسطة فصلات ، وهناك تناظر واحد – إلى – واحد مابين المتغيرات والثوابت .

رغم أن جملة DATA لها نفس تأثير جمل التخصيص إلا أن هناك فرقاً واضحاً وهو أن جملة DATA هي جملة إعلانية غير منفذة ، لذا تتحدد قيم المتغيرات أثناء الترجمة .

من ناحية أخرى ، جمل التخصيص جمل منفذة من ثم ، سيتم القيام بها أثناء التنفيذ .

ملاحظة : أننا نؤكد في مناقشتنا على « إعطاء قيمة إبتدائية » . رغم أننا يمكن أن نعطى قيمة إبتدائية لعداد KOUNT بواسطة :

```
DATA KOUNT/1/
```

إلا أنه يمكن تغير KOUNT أثناء التنفيذ . ولكن حيث أن جملة DATA غير منفذة ، فلا يمكن استدعاؤها أثناء التنفيذ لإعادة إعطاء قيمة إبتدائية للمتغير . وبالتحديد سيكون للعداد KOUNT القيمة 1 طالما لم تخصص قيمة أخرى للعداد KOUNT أثناء التنفيذ ، ولكن بمجرد تخصيص قيمة أخرى للعداد KOUNT لايمكن تنفيذ جملة DATA لإعطاء KOUNT القيمة الإبتدائية 1 مرة أخرى .

يمكن أن نعطى قيم إبتدائية لعناصر متراصة باستخدام جملة DATA . وبالتحديد ، يمكن أن نعطى قيمة إبتدائية لمجموعة متراصة بكتابة إسمها فقط ، فمثلا :

```
DIMENSION GRADE(4)
DATA GRADE/92.0, 78.0, 43.5, 88.0/
```

تخزن 92.0 و ٧٨.0 و 43.5 و 88.0 في GRADE(1) و . . . و GRADE(4) على الترتيب . ونؤكد أننا إذا ذكرنا اسم المجموعة المتراصة ، فسوف تعطى المجموعة المتراصة بأكلها قيما إبتدائية بالترتيب الذى خزنت به المجموعة المتراصة ، أى يتغير الدليل الأول أسرع ، الخ (أنظر قسم ٦ – ٦) . تسمح بعض المترجمات أيضاً باستخدام حلقة DO ضمنية في جملة DATA لإعطاء قيمة إبتدائية لجزء من المجموعة المتراصة فمثلا الجمل التالية :

```
INTEGER A(10)
DATA A(2)/5/, (A(I), I = 4, 7)/1, 2, 3, 4/
```

تخزن القيمة 5 في A(2) والقيم 1 و 2 و 3 و 4 في A(4) و A(5) و A(6) و A(7) على الترتيب .

يسمح الفورتران أيضاً باستخدام معامل تكرار في جملة DATA . وبالتحديد تكتب الرمز .

*n**

قبل أى ثابت حيث *n* هى عدد صحيح موجب بدون إشارة وتدل على أن الثابت سيتكرر عدد *n* من المرات فمثلا :

```
DIMENSION X(100)
DATA X/100*0.0/
```

تدل على أن القيمة 0.0 تتكرر 100 مرة ، ولذا تخصص 0.0 إلى كل عناصر المجموعة المتراصة X وعددها 100 .

يمكن أيضاً استخدام جمل DATA لتخزين ثوابت لسلاسل حرفية وثوابت منطقية (نقصد هنا بثابت سلسلة حرفية ، سلسلة حروف محاطة بفصلات عليا ، مثلا "AVERAGE" و "THE END") .

لايسمح الفورتران القياسى لثوابت السلسلة الحرفية أن تظهر في جمل تخصيص . وبالتالى ، يجب أن نستخدم جملة READ لتخزين حروف بدلا من استخدام جملة DATA . وقد ناقشنا ذلك في قسم ٩ – ٢ . نلخص قواعدها كالآتى :

يتغير الحد الأقصى M من الحروف الذى يمكن تخزينه في أى مكان ذاكرة (ويسمى سعة الحرف) من آلة إلى أخرى. إفرض أن السعة M = 4 من ثم فإن :

```
DATA NAME/'PAT'/
```

تخزن سلسلة الحروف PAT مضبطة من جهة اليسار فى المكان المسمى NAME كالتالى :

NAME	P	A	T	b

مع إضافة مسافة واحدة إلى اليمين . فى حالة السعة M = 2 فإنه يخزن فى الذاكرة الحرفان اللذان على أقصى اليسار PA فقط .

NAME	P	A

وستعتبرها بعض المترجمات مثل WATFOR-WATFIV خطأ طالما تجاوزت طول سلسلة الحروف I السعة M . قد تخزن بعض المترجمات الحروف الزائدة فى المكان التالى من المجموعة المتراصة إذا استخدمنا إسم مجموعة متراصة لإدخال الحروف (أنظر مسألة ١٠-٣) .

يمكننا ببساطة كتابة مايلى لتخزين قيم منطقية باستخدام جمل DATA :

```
LOGICAL A, B
DATA A, B/.TRUE., .FALSE./
```

وكما تمت مناقشته فى قسم ٩ — ٨ .

مثال ١٠ — ١

(أ) إدرس جزء البرنامج التالى :

```
DIMENSION B(100)
LOGICAL L(5)
DATA B/40*0.0, 30*1.0, 20*2.0, 10*3.0/, L/3*.TRUE., .FALSE., .TRUE./
```

من ثم تخصص 0.0 إلى كل من B(1) إلى B(40) وتخصص 1.0 إلى كل من B(41) إلى B(70) وهكذا . يخصص أيضاً TRUE إلى L(1) حتى L(3) و L(5) ولكن تخصص FALSE. إلى L(4) .

(ب) ستقوم الجمل التالية :

```
DIMENSION A(3, 2)
DATA A, B, C, D/1., 2., 3., 4., 5., 6., 3*9.9/
```

بعمل الآتى:

A(1, 1) = 1., A(2, 1) = 2., A(3, 1) = 3., A(1, 2) = 4.,
A(2, 2) = 5., A(3, 2) = 6.   and   B = 9.9, C = 9.9, D = 9.9

١٠ — ٣ حقل —T

يمكن أن تخبر الحاسب بمكان عمود الكتابة أو القراءة بنفس طريقة استخدام مفتاح tab على الآلة الكاتبة . وينفذ ذلك باستخدام حقل —T والشكل العام لمواصفات الحقل هو :

Tp

حيث *p* هى ثابت صحيح بدون إشارة . وتحدد *p* مكان عمود البداية التى سوف تقرأ منها أو تطبع إليها المعلومات .

إفرض ، مثلا ، إننا نريد أن نخزن فى المكانين I ، X القيم المثقبة فى الأعمدة من 13 إلى 15 وفى الأعمدة من 21 إلى 28 على الترتيب فيمكن أن ننجز ذلك بالتالى :

```
      READ(5, 80) I, X
80    FORMAT(12X, I3, 5X, F8.2)
```

يمكن باستخدام حقول —T أن نستخدم FORMAT المكافئة الآتية :

```
80  FORMAT(T13, I3, T21, F8.2)
```

لاحظ أن حقل —T يحدد مكان البداية لمواصفات الحقل التالية .

أحياناً ، نجد أن استخدام حقل —T أبسط حيث أنه يجنبنا عد أماكن الأعمدة .

وحيث أن كل مواصفات الحقول تشير إلى حقول متصلة بزوج الجمل READ-FORMAT .

```
    READ(5, 22) A, B
22  FORMAT(T20, F8.2, 12X, F8.2)
```

تدل ضمناً أن قيمة A ستوجد فى الأعمدة من 20 إلى 27 وقيمة B ستكون فى الأعمدة من 40 إلى 47 .

يمكن قراءة القيم باستخدام حقول —T بأى ترتيب . فمثلا :

```
    READ(5, 23) X, Y
23  FORMAT(T31, F8.2, T11, F8.2)
```

ستخصص إلى X العدد الموجود فى الأعمدة من 31 إلى 38 وتخصص إلى Y العدد الموجود فى الأعمدة من 11 إلى 18

بل يمكننا أيضاً أن نقرأ عنصر عدة مرات . فمثلا :

```
    READ(5, 24) M, N, N1, N2, N3
24  FORMAT(T21, A3, T21, I3, T21, 3I1)
```

ستخزن المعلومات الموجودة فى الأعمدة من 21 إلى 23 كسلسلة حرفية (أى ، فى الشكل الأبجدى الرقى) فى المكان M وكقيمة صحيحة (أى فى الشكل الرقى) فى المكان N والخانات فى الأماكن N_1 و N_2 و N_3 على الترتيب .

فى الخرج ، يعطى حقل —T مكان البداية فى سجل الإخراج (صف) وليس على صفحة الطباعة . وحيث أن الحرف الأول فى سجل الخرج يستخدم للتحكم فى العربة فسيكون مكان البداية على صفحة الطباعة أقل بواحد من الرقم الذى يظهر فى مواصفات حقل —T فمثلا

```
    WRITE(6, 28) I, J, K
28  FORMAT(15X, I7, T28, I5, T42, I9)
```

تأمر الحاسب أن يطبع I مضبطة من اليمين فى الأعمدة من 14 إلى 20 وأن يطبع J مضبطة من اليمين فى الأعمدة من 27 إلى 31 ويطبع K مضبطة من اليمين فى الأعمدة من 41 إلى 49 وهذه مكافئة لما يلى :

```
    WRITE(6, 28) I, J, K
28  FORMAT(T15, I7, 6X, I5, 9X, I9)
```

ومن الطبيعى أنه يمكن أن نطبع القيم بأى ترتيب . فمثلا :

```
    WRITE(6, 39) X, Y
39  FORMAT(T31, F8.2, T11, F8.2)
```

ستطبع Y فى الأعمدة من 10 إلى 17 و X فى الأعمدة من 30 إلى 37 .

١٠ — ٤ حقل —G

يمكن أن يستخدم الشكل المعمم لحقل —G لإدخال / وإخراج أما بيانات صحيحة أو حقيقية أو منطقية ، أومركبة (سوف نناقش البيانات المركبة فى الفصل الحادى عشر) الشكل العام لمواصفات الحقل هو :

Gw.d

حيث تشير w إلى عرض الحقل و d ثابت صحيح بدون إشارة . (يمكن أن نستخدم أيضاً $rGw.d$ حيث تشير r إلى عدد مرات التكرار)
يتحدد معنى $Gw.d$ بواسطة نوع المتغير المناظر كالآتى :

متغيـــر صحيح

$Gw.d$ لها نفس معنى Iw فى كل من المدخل والمخرج .

متغير منطــق :

$Gw.d$ لها نفس معنى Lw فى كل من المدخل وا لخرج .

متغير حقيــق

هناك حالتان :

١ – الإدخال $Gw.d$ لها نفس معنى $Ew.d$ أو $Fw.d$ تبعاً ما إذا كانت القيمة بها E مثقبة فى الحقل الخاص بها أم لا .

٢ – الخرج سترمز d فى هذه الحالة دائماً إلى عدد الخانات المعنوية وسيكون الخرج فى شكل —F إذا أمكن طباعة القيمة فى حقل بعرض 4 — w . فى هذه الحالة ، ستطبع القيمة مضافاً إليها أربع مسافات فقط على الطرف الأيمن الحقل (أى ، مضبطة من اليمين إلى خامس عمود على يمين الحقل) . سيكون الخرج فى شكل —E إذا لم يمكن طباعة القيمة فى حقل بعرض 4 — w . فى هذه الحالة ، ستطبع القيمة مضبطة من اليمين فى الحقل الخاص بها .

متغيـــر مركب

ستتم مناقشته فى الفصل الحادى عشر .

مثال ١٠ – ٢

(أ) إفرض أن I و J و A و B تخزن باستخدام ما يلى .

```
      LOGICAL J
      READ(5, 30) I, J, A, B
   30 FORMAT(3G10.3, G12.3)
```

حيث تثقب بطاقة البيانات كالتالى :

```
         1         2         3         4
123456789012345678901234567890123456789012345
          123  THE END    4.56789  -12.3456E-06
```

وحيث أن G10.3 لها معامل التكرار 3 فتكون مواصفات الحقول G10.3 و G10.3 و G10.3 و G12.3 بعرض 10 ، 10 و 10 و 12 على الترتيب .

(١) تناظر أول G10.3 المتغير الصحيح I لذا فسوف تفسر مثل I 10 من ثم ، تخصص 123 إلى المتغير I وهو الرقم الصحيح المثقب فى الحقل الأول فى الأعمدة من 1 إلى 10 .

(٢) تناظر ثان G10.3 المتغير المنطق J ولذا تفسر مثل L10 وحيث أن أول حرف غير المسافات الخالية فى الحقل الثانى وفى الأعمدة من 11 إلى 20 هو T فسوف يتم تخزين . TRUE . فى J .

(٣) تناظر ثالث G10.3 المتغير الحقيقى A . وحيث لايوجد E مثقبة فى الحقل ، أى فى الأعمدة من 12 إلى 30 لذا فإنها نفس معنى F10.3 ، من ثم يخصص إلى A القيمة 4.56789 .

(٤) تناظر G12.3 المتغير الحقيقى B وحيث أن هناك E مثقبة فى الحقل أى فى الأعمدة من 31 إلى 42 لذا فتصبح E12.3 . من ثم ، يخزن 06 — 12.3456E— أو المكافىء لها 0.0000123456 — فى B

(ب) إفرض أن أمر الإخراج هو :

```
     DO 100 K = 1, 6
            WRITE(6, 30) A(K)
 30         FORMAT(6X, G12.4)
100  CONTINUE
```

حيث A مجموعة متراصة خطية بها ستة عناصر تحتوى ما يلى على الترتيب :

77.777, 0.333×10^{-5}, 2.2222×10^{2}, 666.66×10^{7}, 12.3, 8.88×10^{-3}

الكود G12.4 يقرب كل رقم إلى أربع خانات معنوية :

77.78, 0.00000333, 222.2, 6667000000., 12.30, 0.00888

حيث أن عرض حقل الخرج (الأعمدة من 6 إلى 17) هو 12. فتطبع الأرقام التى لايمكن طباعتها بأربع مسافات خالية على اليمين ، أى الأرقام التى تتطلب أكثر من ٨ = ٤ — ١٢ أعمدة ، فى الشكل الاسى — وبذلك ، تطبع A(2) و A(4) فى الشكل الأسى ، ويظهر الخرج كما فى الشكل ١٠ — ١ . لاحظ أننا نستطيع أن نتعرف على الفور على الأرقام فى الشكل الأسى .



شكل ١٠ — ١

١٠ — ٥ معامل التدريج

عند التعامل مع مجموعات ضخمة من البيانات ، أحياناً يكون من المفيد أن نضمن فى حقول E أو F معامل تدريج ومعامل التدريج له الشكل :

sP

حيث تسمى s (معامل التدريج) وهى ثابت صحيح بدون إشارة أو ثابت صحيح سالب . تكتب على يسار الحقل الأول الذى سوف تطبق عليه ، فمثلا :

3PF8.2 or −4PE14.4

ومع ذلك يختلف ، تأثير معامل التدريج على حقل —F عن تأثيره على حقل —E .

(أ) استخدام معامل التدريج مع حقل —F

عند استخدام معامل التدريج مع حقل —F يكون تأثير *sP* كما يلي :

القيمة الخارجية = القيمة الداخلية × 10^s

أو

القيمة الداخلية = القيمة الخارجية × 10^{-s}

هذا صحيح فى كلا من الإدخال / الاخراج .

مثال ١٠ — ٣

(أ) إفرض أن X تحتوى على الرقم 0.325 ونريد أن نطبع X كنسبة مئوية . يمكن أن نكتب :

```
   WRITE(6, 10) X
10 FORMAT(1X, 2PF8.2, 1X, 'PERCENT')
```

نضرب القيمة الداخلية فى 10^2 ولذا يكون الخرج فى الصورة :

```
32.50 PERCENT
```

(ب) إفرض أن الرقم المثقب فى الأعمدة من 1 إلى 8 من بطاقة بيانات هو 7.50 وعندما تنفذ :

```
   READ(5, 20) RATE
20 FORMAT(2PF8.2)
```

يكون الرقم المخصص إلى RATE هو 0.075 .

وبمعنى آخر ، عندما تستخدم *sP* مع حقل —F فى الإدخال ، سيكون الرقم المخزن هو الرقم المثقب مضروباً فى المعامل 10^{-s} . ولكن عندما نستخدم مع الخرج فسيكون الرقم المطبوع هو الرقم الداخلى مضروباً فى المعامل 10^s .

(ب) استخدام معامل التدريج مع حقل — E

عندما تستخدم معاملات التدريج مع حقول — E فى الإدخال فإنه يتم إهمالها فى الخرج ، لايتغير حجم الرقم المطبوع ، ولكن يضرب الجزء الأساسى للصورة الأسية فى معامل 10^s ويخفض الأس بمقدار *s* . (تذكر أن الجزء الأساسى للصورة الأسية هو رقم مابين 0.1 و 1.0 أو 0.1 — و 1.0 —) .

فمثلا ، إفرض أن القيمة الداخلية X هى 0.0004321 ستظهر X إذا طبعت باستخدام مواصفات الحقل E12.4 ، فى الشكل :

```
0.4321E – 03
```

من ناحية أخرى ، ستظهر X إذا طبعت باستخدام مواصفات الحقل 1PE12.4 ، فى الشكل

```
4.321E – 04
```

(الآن تظهر قيمة X فى الترميز العلمى ، أى ، كرقم مابين 1.0 و 10.0 أو 1.0 — و 10.0 — متبوعة بقوة 10) .

تحذير : بمجرد استخدام معامل التدريج فى حقل E أو F لأول مرة فإنه يطبق أوتوماتيكياً على جميع حقول E ، F التالية فى البرنامج إلى أن يقابل معامل آخر . لذلك ، يجب أن نستخدم معامل التدريج

0P (zeroP)

عندما نريد أن نلغى تأثير معامل تدريج سابق . فمثلا ، إفرض أن قيم A و B و C و D الداخلية هى 0.0111 و 0.0222 و 0.0333 و 0.0444 على الترتيب وأوامر الخرج هى كما يلى :

```
      WRITE(6, 10) A, B
10    FORM/ T(1X, 1PE12.4, 5X, E12.4)
      WRITE(6, 20) C, D
20    FORMAT(1X, E12.4, 5X, 0PE12.4)
```

فسوف يطبق معامل التدريج 1P لطباعة A و B و C ولكن لايطبق على D أو أى قيمة حقيقية ؤ أى جملة WRITE تالية ، وبذلك سيظهر الخرج فى الشكل :

```
1.110E-02     2.220E-02
3.330E-02     0.4440E-01
```

١٠ ـ ٦ قاعدة الاقواس اليسرى

تذكر أنه إذا احتوت قائمة المتغيرات فى جملة READ أو WRITE على عناصر أكثر من تلك التى فى مواصفات الحقول فى جملة FORMAT المصاحبة فإن جملة FORMAT تكرر عدة مرات (تقرأ بطاقة بيانات جديدة أو يطبع سطر جديد فى كل مرة) إلى أن يتم الانتهاء من كل المتغيرات فثلا افرض أننا نفذنا الآتى :

```
      READ(5, 10) A, B, C, D, E, F, G, H
10    FORMAT(F6.2, F8.3, F5.1)
```

فسوف نستخدم جملة FORMAT لقراءة قيم A و B و C (على أول بطاقة بيانات) وبعد ذلك تكرر جملة FORMAT لـ D و E و F (على ثانى بطاقة بيانات) ، وبعد ذلك تكرر لكل من G و H (على ثالث بطاقة بيانات) .

قد ترغب أحياناً فى تكرار جزء فقط من جملة FORMAT والقاعدة هى تكرار جملة FORMAT ابتداء من القوس الأيسر إلى أقصى اليمين وحتى نهاية جملة FORMAT .

فثلا ، إفرض أننا قمنا بتعديل جملة FORMAT السابقة كالتالى :

```
      READ(5, 20) A, B, C, D, E, F, G, H
20    FORMAT(F6.2, (F8.3, F5.1))
```

بعد استخدام جملة FORMAT لـ A و B و C تكرر جملة FORMAT ولكن إبتداء من القوس الأيسر أقصى اليمين فقط ، أى ، إبتداء من مواصفات الحقل F8.3 من ثم ، تقرأ D و E من بطاقة البيانات الثانية و F و G من بطاقة البيانات الثالثة و H من بطاقة البيانات الرابعة . وكل مرة باستخدام المواصفات F8.3 و F5.1 .

تطبق القاعدة السابقة أيضاً على الأقواس التى تسبق بمعامل تكرار . نجد مثلا

```
      READ(5, 30) A, B, C, D, E, F, G, H
30    FORMAT(F6.2, 2(F8.3, 3X), F5.1)
```

تكافىء جملة FORMAT الخاصة بالجملة التالية :

```
30    FORMAT(F6.2, (F8.3, 3X, F8.3, 3X), F5.1)
```

بعد استخدام جملة FORMAT لـ A و B و C و D تكرر جملة FORMAT إبتداء من القوس الأيسر فى أقصى اليمين ، أى ، من F8.3 إلى نهاية الجملة (وليس إلى نهاية القوس الداخل) . وبذلك تخصص قيم إلى E و F و G من ثانى بطاقة بيانات باستخدام F8.3 و 3X و F8.3 و 3X و F5.1 وتخصص قيمة إلى H من ثالث بطاقة بيانات باستخدام F8.3 .

تطبق القاعدة السابقة أيضاً على جمل FORMAT التى تصاحب جمل WRITE .

١٠ — ٧ صيغ (FORMAT) وقت التنفيذ

تبعاً للملامح التي تمت مناقشتها حتى الآن ، فإنه بمجرد كتابة البرنامج فسوف يتم تحديد كل جمل FORMAT وبالتالي تكون ثابتة وقد يكون من الملائم جداً إذا أمكن تعديل جمل FORMAT أثناء التنفيذ لتناسب البيانات ، فمثلا ، قد نحتاج أن نكبر عرض الحقول إذا كانت القيم أكبر مما توقعنا أصلا . يناقش هذا القسم هذه المقدرة باستخدام مايسمى بجمل صيغ (FORMAT) وقت التنفيذ (ويحددها المستفيد) .

وكما يشير الإسم فجملة FORMAT التي تعطى (تحدد) أثناء التنفيذ تسمى بجمل صيغة (FORMAT) وقت التنفيذ . والفكرة العامة وراء صيغة (FORMAT) وقت التنفيذ هي تخزين (إدخال) مواصفات الحقول كسلسلة حروف فى مجموعة متراصة . وبعد ذلك تستدعى جملة WRITE/READ المجموعة المتراصة إذا أردنا استخدام مواصفات الحقول فى المجموعة المتراصة للإدخال / الإخراج .

إدرس زوج الجمل READ-FORMAT التالية :

```
    READ(5, 10) A, B, I, J
10  FORMAT(2(F10.2, 2X), I5, 2X, I8)
```

بفرض أن سعة الحرف M = 4 ، نوضح كيف يمكن أن نحصل على نفس النتيجة باستخدام صيغ (FORMAT) وقت التنفيذ .

أولا نثقب مايلى :

```
(2(F10.2, 2X), I5, 2X, I8)
```

على بطاقة بيانات ، وبعد ذلك نقرأ سلسلة الحروف (باستخدام حقل A) فى مجموعة متراصة . وليكن :

```
    INTEGER FORM(6)
        ⋮
    READ(5, 20) FORM
20  FORMAT(6A4)
```

الحروف المخزنة فى المجموعة المتراصة FORM هى كالآتى :

FORM(1)	(	2	(	F
FORM(2)	1	0	.	2
FORM(3)	,	2	X	)
FORM(4)	,	I	5	,
FORM(5)	2	X	,	I
FORM(6)	8	)		

لاستخدام صيغ (FORMAT) وقت التنفيذ هذه ، تكتب ما يلى :

```
READ(5, FORM) A, B, I, J
```

لاحظ أن اسم المجموعة المتراصة FORM (مكان تخزين مواصفات الحقول) يكتب بدلا من رقم جملة FORMAT .

يمكن أن يكون المجموعة المتراصة المستخدمة لمواصفات الأشكال أى عدد من العناصر – طالما أنها كبيرة بدرجة كافية (لت المواصفات المتوقعة) . وغالباً يستخدم ما يلى للمرونة الكاملة :

```
    INTEGER FORM(80)
    READ(5, 40) FORM
40  FORMAT(80A1)
```

وهذا يلائم أى موصفات مثقبة على بطاقة بيانات واحدة وبأى سعة M .

ونلخص ماسبق بأصدار قائمة النقاط الهامة التى يجب أن نتذكرها :

١ – يجب أن تخزن الصيغة (FORMAT) المطلوبة فى مجموعة متراصة حتى يمكن تخزينها فى مكان واحد فقط ، فمثلا رغم أن :

(1X, I8)

يمكن تخزينها فى مكان واحد INFOR عندما تكون سعة الحروف M = 9 ، ولكن لزاماً علينا أن نعلن أن INFOR مجموعة متراصة .

٢ – عند تحضير بطاقة البيانات لأكواد الإشكال لا تثقب كلمة FORMAT ولكن تثقب مواصفات الحقول بداخل أقواس .

٣ – عند استخدام صيغة (FORMAT) محددة بالمستفيد ، يظهر إسم المجموعة المتراصة المحتوى على مواصفات الحقول فى جملة WRITE/READ فمثلا ،

```
WRITE(6, INFOR) A, B, C, J, K
```

تخبر الحاسب أن مواصفات الأشكال مخزنة فى المجموعة المتراصة INFOR

بدلا من قراءة مواصفات الأشكال فى مجموعة متراصة كما سبق ، يمكن أيضاً أن نعطى قيمة إبتدائية لصيغة (FORMAT) وقت التنفيذ باستخدام جملة DATA :

```
INTEGER FORM(3)
DATA FORM/'(F8.', '3,I8', ')'/
```

إذا سمح المترجم بظهور ثوابت كسلسلة حرفية فى جمل التخصيص ، فيمكننا أن نستخدم :

```
INTEGER FORM(3)
FORM(1) = '(F8.'
FORM(2) = '3,I8'
FORM(3) = ')'
```

(ومن الطبيعى ، فن الممكن استخدام حقل هولوريث أيضاً) .

بمجرد تخزين مواصفات الصيغة فى مجموعة متراصة كما سبق ، فيمكن أيضاً أن نعدل مواصفات الصيغ وفقاً البيانات الناتجة . توضح هنا هذا الأسلوب الفنى بأمثلة ،

١ – افرض أنه تم تحديد الصيغة مسبقاً بواسطة مايلى :

```
DIMENSION INFOR(3), X(100)
DATA INFOR/'(I3,', '2X,F', '6.2)'/, IBIG/'8.2)'/, ISMALL/'6.2)'/
```

معنى هذا أنه تم تخزين (I 3 و 2X و F6.2) فى INFOR . جزء البرنامج التالى :

```
     DO 100 K = 1, 100
            IF(ABS(X(K)).GT.100.0) INFOR(3) = IBIG
100  CONTINUE
     WRITE(6, INFOR) (N, X(N), N = 1, 100)
```

سيتحقق أولا من أن قيمة أى عنصر فى المجموعة المتراصة X يتجاوز الرقم 100 . إن كان كذلك ، فتتغير مواصفات الصيغة F6.2 إلى F8.2 وذلك للسماح بمساحة أكبر تلائم البيانات . وبالتالى فإن :

```
      DO 200 K = 1, 100
         IF(ABS(X(K)).GT.100.0) INFOR(3) = IBIG
         WRITE(6, INFOR) K, X(K)
         INFOR(3) = ISMALL
  200 CONTINUE
```

تتغير F6.2 إلى F8.2 فقط بالنسبة لعناصر معينة فى المجموعة المتراصة والتى تتجاوز قيمتها الرقم 100 . تبقى F6.2 لكل العناصر الباقية .

٢ - إفرض أننا نفضل حقل F لطباعة المجموعة المتراصة A وقد تم استخدام F12.4 . ومع ذلك كلما كبرت قيمة A(K) بحيث لاتناسب 12 عمود ، فإن الطبع بصورة شكل E يكون مقبولا , جزء البرنامج التالى يقوم بإجراء ذلك تماماً :

```
      INTEGER XFORM, EFORM, FFORM
      DIMENSION A(100), XFORM(4)
      DATA XFORM/'(1X,', 'I3,2', 'X,F1', '2.4)'/, EFORM/'X,E1'/, FFORM/'X,F1'/
            :
      DO 300 K = 1, 100
         IF(ABS(A(K)).GT.1.0E6) XFORM(3) = EFORM
         WRITE(6, XFORM) K, A(K)
         XFORM(3) = FFORM
  300 CONTINUE
```

لاحظ أن مواصفات الصيغة الأصلية هى (F12.4 و 2X و I 3 و 1 X) ولكن تم تغيير F إلى E كلما كانت A(K) كبيرة جداً ، أى ، كلما كبرت قيمتها عن 10^6 .

١٠ - ٨ الرسم البيانى

إلى جانب استخدام الحاسب لحساب قوائم من القيم . فباستطاعتنا أيضاً أن نستخدم الحاسب ليرسم القيم فى شكل بيانى . سيتكون الرسم البيانى من خطوط من النقط « ...... » للمحورين وحرف "X" لكل نقطة (x, y) محسوبة . وبرسم عدد كاف من النقط نحصل على صورة الرسم البيانى . ونوضح فيما بعد هذا الأسلوب الفنى .

إفرض ، مثلا أننا نريد أن نرسم الدالة :

$$y = 2x^3 - x^2 - 22x + 21$$

لكل من $-4 \le x \le 4$ أولا يجب أن نختار قيم لـ x بين -4 ، 4 وذلك لحساب y وليكن :

$$x = -4, -3.9, -3.8, \ldots, 3.9, 4$$

لاحظ أن هناك 81 قيمة لـ x وأن معامل الزيادة 0.1 . ولكننا نحتاج أيضاً مدى y لهذه القيم . إفرض أنه $-50 \le y \le 50$ تناظر $-4 \le x \le 4$ بعد ذلك نجعل كل مكان على محور y يدل على القيم المتزايدة لـ y . إذا اخترنا معامل الزيادة 1.0 ، فالأماكن على محور y تمثل القيم :

$$y = -50, -49, -48, \ldots, 49, 50$$

لاحظ أن هناك 101 قيمة لـ y .

هناك طريقتان لاستكمال المسألة .

الطريقـة الأولى :

نختار المحور الأفقى (بعرض صفحة الطباعة) كمحور x والمحور الرأسى (بطول صفحة الطباعة) كمحور y . فى هذه الحالة ، يجب أن نحسب كل القيم (x, y) ونخزن الرسم البيانى بأكمله فى مصفوفة G قبل أن نتمكن من تنفيذ الرسم البيانى . السبب فى ذلك هو أن وحدة الطباعة تطبع سطراً واحداً فى المرة ، ومن الممكن أن يكون لقيمتين مختلفتين لـ x نفس قيمة y (الأفقية) . (ليس فى الإمكان أن نحرك العربة أعلى وأسفل صفحة الطباعة) وبذلك سيكون لـ G 101 صف تناظر أماكن y الـ 101 و 81 عموداً تناظر 81 قيمة لـ x أى أن G ستكون مصفوفة (81 × 101) . أسوأ ما فى هذه الطريقة هو أن الرسم البيانى يحتاج مصفوفة بها 8181 = 81 × 101 خلية ذاكرة ولهذا السبب فعادة ترسم الرسوم البيانية باستخدام الطريقة الثانية .

الطريقـة الثـانية :

نختار المحور الأفقى كمحور y والمحور الرأسى كمحور x (يستطيع القارىء أن يقلب الصفحة ليتابع الرسم) حيث أن كل قيمة لـ x تعطى قيمة فريدة لـ y ، نستطيع أن نحسب ونرسم رسمنا البيانى سطر واحد فى المرة . بالتالى ، سنحتاج فقط لمجموعة متراصة خطية بها 101 خلية .

بالتحديد نتبع الخطوات التالية :

١ – إجعل قيمة x مساوية لقيمتها الصغرى .

٢ – أوجد القيمة المناظرة y .

٣ – استخدم مقياس رسم لقيمة y رقم صحيح J يتراوح بين 1 و 101

٤ – اطبع سطراً به حرف "X" فى العمود J من الرسم البيانى .

٥ – زد قيمة x وكرر الخطوات من ١ إلى ٤ طالما لم تتجاوز x قيمتها العظمى .

الخطوات السابقة واضحة المعالم مباشرة . وفى الخطوة الرابعة نريد أيضاً أن نطبع نقطة (.) للعمود الذى يمثل محور x (عند $y = 0$) ونريد أيضاً أن نطبع سطراً كاملا من النقط (....) عندما يمثل السطر محور y (عند $x = 0$) . هذه المهام مفصلة ولكنها ليست صعبة .

نحصل على دالة مقياس الرسم فى الخطوة ٣ كما يلى . من المعلوم أن y تقع ما بين 50 — و 50 بجمع 51 على y تنتج رقماً حقيقياً موجباً بين 1 و 101 . حيث نريد تقريب y إلى رقم صحيح بين 1 و 101 ، بدلا من البتر ، يجب أيضاً أن نجمع 0.5 إلى y . يعطى هذا دالة مقياس الرسم :

```
JSCALE(Y) = INT(Y + 51.0 + 0.5)
```

لاحظ أن JSCALE(0) = 51 أى أن عمود 51 من الرسم البيانى هو محور x .

وفيما يلي البرنامج الذي يرسم دالتنا :

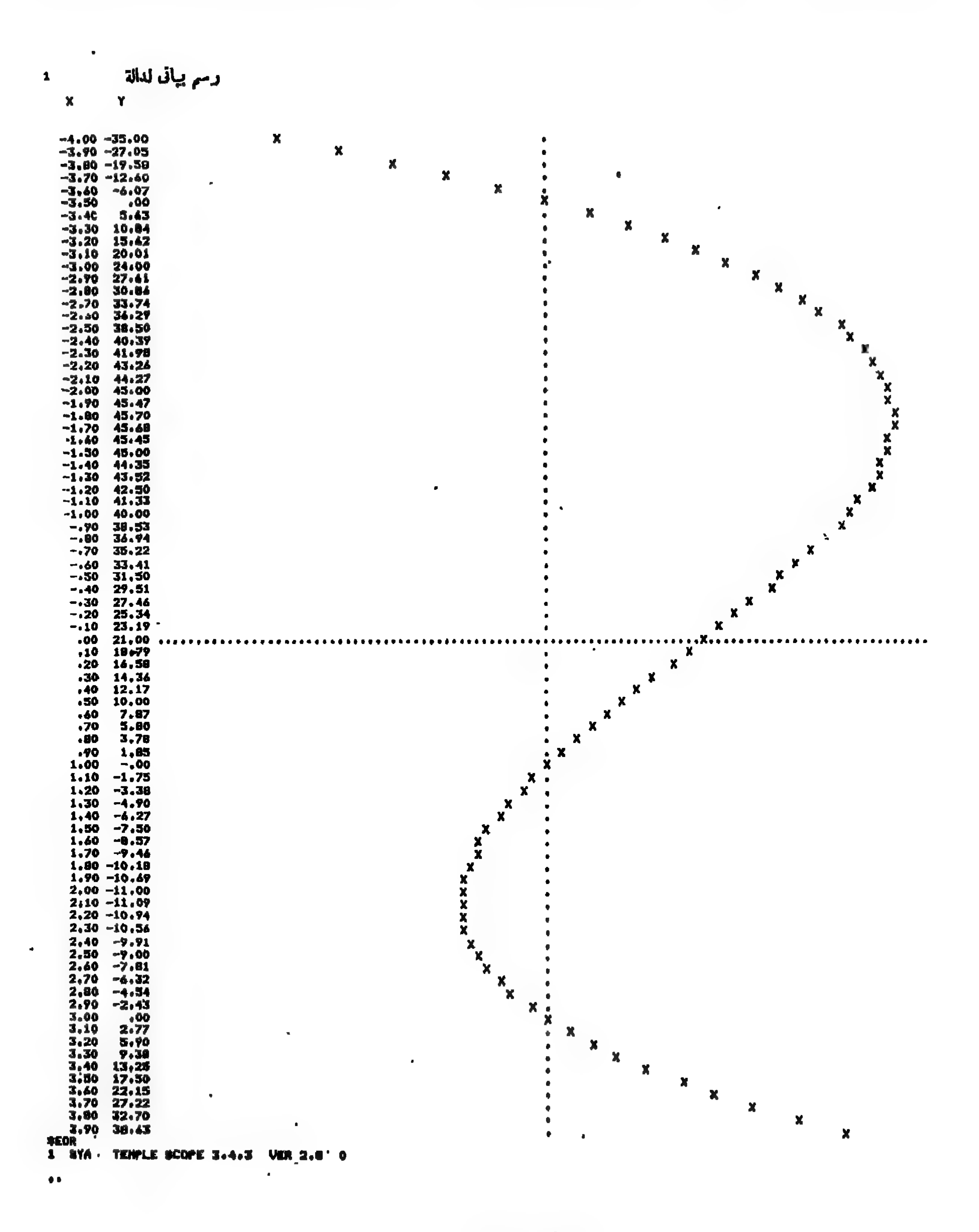
```
      INTEGER BLANK, DOT, CROSS, LINE(101), LINEY(101)
      DATA BLANK, DOT, CROSS, LINE, LINEY/' ', '.', 'X', 101*' ', 101*'.'/
C
C     BEGIN NEW PAGE AND SKIP LINES
C
      WRITE(6, 10)
   10 FORMAT('1', 10X, 'GRAPH OF A FUNCTION'//3X, 'X', 6X, 'Y'//)
C
C     BEGIN GIVING VALUES TO X
C
      X = -4.0
   50 Y = ((2.0*X - 1.0)*X - 22.0)*X + 21.0
      J  = INT(Y + 51.5)
C             TEST IF Y AXIS
      IF(ABS(X).LT.0.001) GO TO 100
      LINE(51) = DOT
      LINE(J) = CROSS
      WRITE(6, 20) X, Y, LINE
   20 FORMAT(1X, 2(F6.2, 1X), 101A1)
      LINE(J) = BLANK
      GO TO 200
  100 LINEY(J) = CROSS
      WRITE(6, 20) X, Y, LINEY
  200 X = X + 0.1
      IF(X.LE.4.0) GO TO 50
      STOP
      END
```

لاحظ أننا استخدمنا جملة DATA لتخزين الحروف b (مسافة) ، . (نقطة) و X (حرف X) في المتغيرات BLANK و DOT وCROSS على الترتيب . تستخدم LINE لطباعة كل سطر من الرسم البياني إلا عندما يمثل السطر محور y . في هذه الحالة تستخدم LINE Y. من ثم ، يعطى LINE قيمة إبتدائية « مسافات » ولكن LINE Y تعطى قيمة إبتدائية « نقط » . (كان يمكننا أن نستخدم مجموعه متراصة واحدة ولكن حينئذ كان يجب أن نعيد مساواة المجموعة المتراصة بمسافات أو بنقط بعد كل زيادة في X) . السطر التالي :

LINE(J) = BLANK

هام فهو يمسح حرف X في LINE (J) بعد طباعته — واننا نذكر أنه يمكن أن نستخدم حلقة DO لقيم X المختلفة . يظهر خرج البرنامج في شكل ١٠ — ٢ لاحظ أن قيم x و y تظهر على الجانب الأيسر من الرسم البياني .

ملحوظة : تتطلب دالة مقياس الرسم JSCALE(Y) التي تمت مناقشتها فيما سبق وفي المسألة ١٠ — ١٦ ، أن نعرف مسبقاً مدى قيم y . في حالة ما إذا لم تكن هذه المعلومات متوافرة لدينا نستطيع أن نواصل كما يلي : (١) احسب كل قيم y وخزنها في مجموعة متراصة Y (قبل عمل أي رسم) . (٢) أوجد القيم العظمى والصغرى في المجموعة المتراصة Y . (٣) استخدم هذه القيم للحصول على دالة مقياس رسم JSCALE كما في المسألة ١٠ — ١٦ . (٤) ارسم الرسم البياني باستخدام الدالة JSCALE والمجموعة المتراصة Y . (تم توصيف هذا الإجراء أيضاً في المسألتين ١٠ — ٣٦ و ١٠ — ٣٧) .



1 رسم بيانى لدالة

X	Y
-4.00	-35.00
-3.90	-27.05
-3.80	-19.58
-3.70	-12.60
-3.60	-6.07
-3.50	.00
-3.40	5.63
-3.30	10.84
-3.20	15.62
-3.10	20.01
-3.00	24.00
-2.90	27.61
-2.80	30.86
-2.70	33.74
-2.60	36.29
-2.50	38.50
-2.40	40.39
-2.30	41.98
-2.20	43.26
-2.10	44.27
-2.00	45.00
-1.90	45.47
-1.80	45.70
-1.70	45.68
-1.60	45.45
-1.50	45.00
-1.40	44.35
-1.30	43.52
-1.20	42.50
-1.10	41.33
-1.00	40.00
-.90	38.53
-.80	36.94
-.70	35.22
-.60	33.41
-.50	31.50
-.40	29.51
-.30	27.46
-.20	25.34
-.10	23.19
.00	21.00
.10	18.79
.20	16.58
.30	14.36
.40	12.17
.50	10.00
.60	7.87
.70	5.80
.80	3.78
.90	1.85
1.00	-.00
1.10	-1.75
1.20	-3.38
1.30	-4.90
1.40	-6.27
1.50	-7.50
1.60	-8.57
1.70	-9.46
1.80	-10.18
1.90	-10.69
2.00	-11.00
2.10	-11.09
2.20	-10.94
2.30	-10.56
2.40	-9.91
2.50	-9.00
2.60	-7.81
2.70	-6.32
2.80	-4.54
2.90	-2.43
3.00	.00
3.10	2.77
3.20	5.90
3.30	9.38
3.40	13.28
3.50	17.50
3.60	22.15
3.70	27.22
3.80	32.70
3.90	38.63

$EOR
1 SYA · TEMPLE SCOPE 3.4.3 VER_2.8' 0

شكل ١٠ – ٢

مســائل محلولة

جمــلة DATA :

١٠ – ١ أوجد قيم المدخلات :

(أ)
```
DATA A, B, C /2.4, '2.4', 1.5/
```
(ب)
```
DATA A, B /2*2.4/, C/1.5/
```
(جـ)
```
DIMENSION A(10)
DATA A, B, C/6*1.0, 6*2.0/
```
(د)
```
LOGICAL C
DATA A, B, C /2.4, 'TRUE', .TRUE./ .
```

(أ) تخصص القيم بين الشرطات المائلة لكل من A و B و C من ثم تخصص 2.4 إلى A تخصص 1.5 إلى C وتخصص سلسلة الحروف التالية إلى B :

B | 2 | . | 4 | b |

(مع فرض أن سعة الحرف M = 4)

(ب) 2* هى معامل التكرار . من ثم تخصص 2.4 إلى كل من A و B وتخصص 1.5 إلى C .

(جـ) 6* هى معامل التكرار . ومن ثم يخصص 1.0 إلى أول ستة عناصر من A وتخصص 2.0 إلى B ، C وإلى آخر أربعة عناصر من A .

(د) تخصص 2.4 إلى A وتخصص سلسلة الحروف .TRUE. ، إلى B وتخصص القيمة المنطقية . TRUE . إلى C .

١٠ – ٢ أوجد المدخلات

```
LOGICAL A(3, 2)
DATA A/2*.TRUE., 3*.FALSE., .TRUE./
```

ترتب المجموعة المتراصة A بحيث يكون الدليل الأول (دليل الصف) هو الذى يتغير أسرع . من ثم تخصص .TRUE. إلى A(1, 1) و A(2, 1) وتخصص .FALSE. إلى (A(3,1 و A(1,2) و A(2,2) وتخصص .TRUE. إلى A(3, 2)

١٠ – ٣ إذا فرضنا أن سعة الحرف M = 4 أوجد البيانات فى المخزن عند تنفيذ مايلى :

```
DIMENSION L(2)
DATA L/'SLEEP', 'IN'/
```

حيث أن طول أول سلسلة حروف l تتجاوز السعة M ، فيمكن حدوث أى مما يلى :

(١) تخزن كل سلسلة حروف مضبطة من اليسار فى الذاكرة مع بتر الحروف الزائدة أو تضاف مسافات زائدة على اليمين .

L(1) | S | L | E | E | L(2) | I | N | b | b |

(٢) تعطى رسالة خطأ .

(٣) حيث أن إسم المجموعة المتراصة L يتم استخدامه فى جملة DATA فإن SLEEP تخزن مضبطة من اليسار فى L(1) وتحرك الحروف الزائدة إلى المكان التالى L(2) كما يلى :

L(1) | S | L | E | E | L(2) | P | I | N | b |

(تؤكد أن كل المترجمات تعطى نفس النتيجة إذا لم يتجاوز طول سلسلة الحروف السمة M) .

حقــل —T

١٠ — ٤ إفرض أنه تم تثقيب بطاقة بيانات كما يلى :

```
                    1                   2                   3
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
1 1 1 1   2 2 2 2   T H E   E N D       3 3 3 3
```

أوجد المدخلات إذا نفذنا الآتى :

(أ)
```
      READ(5, 10) I, J
10    FORMAT(T6, I4, T3, I6)
```

(ب)
```
      LOGICAL J
      READ(5, 20) I, J, X, Y
20    FORMAT(T11, A4, T11, L4, T21, F5.2, T3, F6.3)
```

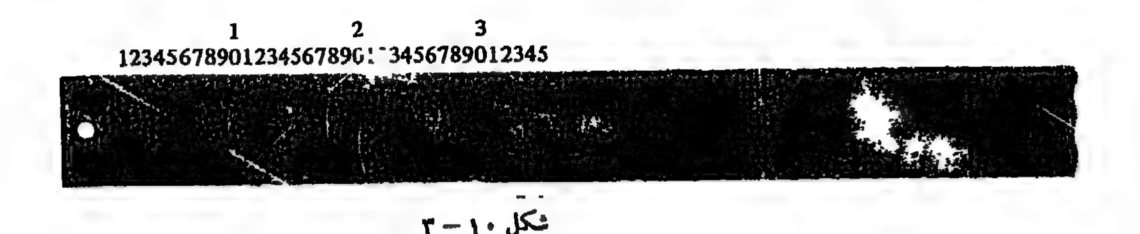
(أ) يخصص الرقم الصحيح الموجود فى الأعمدة من 6 إلى 9 إلى I أى ، الحقل الذى يبدأ فى العمود 6 وبعرض 4 . من ثم فيخصص الرقم 2222 ؛ إلى I ، ويخصص الرقم الصحيح الموجود فى الإعمدة من 3 إلى 8 إلى J أى فى الحقل الذى يبدأ فى العمود 3 وبعرض 6 . وحيث أن المسافات (الفراغات) تفسر كأصفار فى الحقول الرقية ، فتخصص 110222 إلى J .

(ب) تخصص سلسلة الحروف «THE» إلى I وهى الحروف الموجودة فى الحقل الذى يبدأ فى العمود 11 بعرض 4 . تخصص TRUE . إلى J حيث أن J متغير منطقى ، T هى أول حرف غير خال فى الحقل ذى العرض 4 الذى يبدأ فى العمود 11 . (لاحظ أن كلا من I و J يستخدم البيانات المثقبة فى الأعمدة من 1 إلى 4) . وحيث أن T_{21} و T_3 تأمر وحدة قراءة البطاقات بالذهاب إلى عمود 21 ثم 3 على الترتيب ، تخصص 333.30 إلى X وتخصص 110.222 إلى Y .

١٠ — ٥ أوجد الخرج إذا نفذنا البرنامج الآتى :

```
      I = 111
      J = 222
      K = 333
      WRITE(5, 30) I, J, K
30    FORMAT(T15, I3, T6, I6, T20, I8)
```

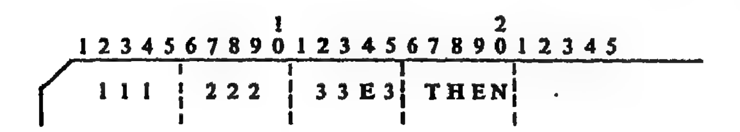
تشير T_p فى الخرج إلى المكان p فى سجل الإخراج (سجل) من ثم تشير T_p إلى العمود رقم $p - 1$ على صفحة الطباعة حيث لا يطبع أول حرف فى السجل . بالتالى ، تطبع I مضبطة من اليمين فى الحقل ذى العرض 3 الذى يبدأ فى العمود 14 على صفحة الطباعة ، أى فى الأعمدة من 14 إلى 16 ؛ تطبع J مضبطة من اليمين فى الحقل ذى العرض 6 الذى يبدأ فى العمود 5 ، أى فى الأعمدة 5 إلى 10 وتطبع K مضبطة من اليمين فى الحقل ذى العرض 8 الذى يبدأ فى العمود 19 أى فى الأعمدة من 19 إلى 26 . أنظر شكل ١٠ — ٣ .



شكل ١٠ – ٣

حقـــل G

١٠ – ٦ إفرض أنه تم تثقيب بطاقة بيانات كالآتى :



أوجد الخرج إذا نفذنا الآتى :

```
      LOGICAL Z
      READ(5, 10) J, A, B, Z
10    FORMAT(4G5.1)
```

هناك 5 مداخل G5.1 ، مدخل لكل متغير . لذا فهناك أربعة حقول كل منها بعرض د .

(أ) حيث أن J متغير صحيح ، لذا تفسر G5.1 مثل I5 . ومن ثم تخصص 1110 إلى J وهو الرقم الصحيح الظاهر فى الأعمدة من 1 إلى 5 (اعتبرنا المسافة الخالية فى عمود 5 صفراً) .

(ب) حيث A متغير حقيقى وليس هناك E مثقبة فى الأعمدة من 6 إلى 10 ومن ثم ، تفسر G5.1 مثل F5.1 . وحيث أن العلامة العشرية غير موجودة فى الحقل ، لذا تضاف علامة عشرية بعد مكان واحد من الجانب الأيمن للحقل ، وبذلك تخصص 222.0 إلى A .

(ﺟ) حيث B متغير حقيقى وحيث أن هناك E مثقبة فى الأعمدة من 11 إلى 15 من ثم تفسر G5.1 مثل E5.1 وحيث أن العلامة العشرية غير موجودة فى الحقل ، لذا تضاف علامة عشرية بعد مكان واحد من الجانب الأيسر لحرف B وبذلك تخصص 3.3E3 أو مكافئها 3300 إلى B .

(د) حيث Z متغير منطق لذا تفسر G5.1 مثل L5 . وحيث أن أول حرف غير خال فى الأعمدة من 16 إلى 20 هو T . بذلك تخصص .TRUE. إلى Z .

١٠ – ٧ إفرض أن J تحتوى على 2345 وأن K متغير منطق يحتوى على . .FALSE . أوجد الخرج إذا نفذنا الآتى :

```
      WRITE(6, 20) J, K
20    FORMAT(1X, 2G10.3)
```

باستبعاد أمر التحكم فى العربة 1X ، فعرض الحقول J و K هى عشرة حروف لكل منهما . وحيث أن J متغير صحيح فإن G10.3 تصبح I 10 ولذا تطبع 2345 مضبطة من اليمين فى الأعمدة من 1 إلى 10 . وحيث أن K متغير منطق ، فإن G10.3 تصبح L10 ، ولذا تطبع F فى العمود 20 .

١٠ – ٨ إفرض أن A مجموعة متراصة خطية مكونة من سبعة عناصر تحتوى على مايلى على الترتيب :

111.18, −22227000.0, −6.15, 0.0000033333, 44.4, −0.023, 888.88

أوجد الخرج التالى :

```
      DO 100 K = 1, 7
            WRITE(6, 40) A(K)
 40         FORMAT(1X, G12.4)
100   CONTINUE
```

تقرب G12.4 كل رقم إلى أربعة أماكن معنوية كما يلى :

111.2, −22230000.0, −6.150, 0.000003333, 44.40, −0.02300, 888.9

سوف تطبع فى الشكل الأسى الأرقام التى لايمكن طباعتها فى حقول ذات عرض $8 = 4 - 12 = 4 - w$ (أى التى تكون قيمتها المطلقة كبيرة أو صغيرة نسبياً) . وسوف تطبع الأرقام الأخرى فى شكل F— مضبطة من اليمين ومنتهية بأربع مسافات كالتالى :

```
  111.2
-0.2223E 08
 -6.150
 0.3333E-05
  44.40
-0.02300
  888.9
```

معامل المقيـــاس التدريجى

١٠ – ٩ أوجد القيمة الداخلية إذا أعطيت أكواد الصيغ الآتية وكذا الرقم الخارجى فى المدخلات .

	أكواد الصيغ		الرقم الخــارجى كدخل
(أ)	3PF6.1	(٢)	123.45
(ب)	−4PF6.1		123.45
(جـ)	4PE6.1		123.45
(د)	−3PE8.1		123.45E1
(هـ)	1PF6.2		12345.
(و)	−1PE9.2		12345E − 2

تأثير معامل التدريج sP على الأرقام الخارجية كدخل مكتوب من غير اس وفيما يلى :

الرقم الداخل = الرقم الخارجى × 10^{-s}

لايؤثر معامل التدرج على الأرقام المكتوبة بأس كما يلى (d) و (f) من ثم ،

(أ) 0.12345 (ب) 1234500 (جـ) 0.012345 (د) 1234.5 أو 123.45E1 (هـ) 1234.5

(و) تضاف علامة عشرية بين 3 ، 4 لتعطى 2 — 123.45E أو 1.2345 . ليس لمعامل التدريج أثر على الرقم .

١٠ – ١٠ إفرض أن 12345.6 هو الرقم الداخل . أوجد الخرج إذا كانت مواصفات الحقل هى :

(أ) 2PF12.1 (جـ) −1PF12.2 (هـ) −1PE15.6
(ب) −3PF12.2 (د) 2PE12.4 (و) 1PE15.3

تذكر أولا أن $Fw.d$ تقرب الرقم إلى عدد d من الأماكن العشرية بينما تقرب $Ew.d$ الرقم إلى عدد d من الأرقام المعنوية .

بالنسبة لحقل F فيضرب معامل التدريج sP القيمة الداخلية فى 10^s من ثم ، تصبح لدينا ما يأتى :

(أ) 1234560.0 (ب) 12.35 (جـ) 1234.56

بالنسبة لحقل E فلا يغير معامل التدريج قيمة الرقم ، كما يحدث مع حقل F ولكن تغير فى الشكل الأسى فقط . وبالتحديد ، تحرك العلامة العشرية بضرب الجزء الأساسى للصورة الأسية فى 10^s ثم تضاف s — إلى الأس ويصبح لدينا مايلى :

( ) 12.35E + 03 (هـ) 0.0123456E + 06 (و) 1.23E + 04

١٠ – ١١ إفرض أن A تحتوى على 111.888 أوجد الخرج إذا كان الأمر هو :

```
      WRITE(6, 10) A, A, A, A, A
   10 FORMAT(1X, F10.2, E12.4, 1PF10.2, E12.4, F10.2)
```

يطبق معامل التدريج sP على مواصفات كل حقل تال حتى يتم مقابلة معامل تدريج آخر . من ثم ، يكون الخرج كما يلى :

111.89 0.1119E + 03 1118.88 1.119E + 02 1118.88

حيث تطبع الأرقام مضبطة من اليمين فى حقول ذات عرض 10 ، 12 ، 10 ، 12 ، 10 على الترتيب .

١٠ – ١٢ إفرض فى المسألة السابقة ١٠ – ١١ ، أن المطلوب هو تطبيق معامل التدريج على A الثالثة وليس على الرابعة والخامسة . كيف نكتب جملة FORMAT ؟

تكتب 0P عندما لانرغب فى تطبيق معامل التدريج على أى حقل تال بعد ذلك . من ثم ستكتب جملة FORMAT كالتالى :

```
   10 FORMAT(1X, F10.2, E12.4, 1PF10.2, 0PE12.4, F10.2)
```

صيغة (FORMAT) وقت التنفيذ :

١٠ – ١٣ باعتبار أن سعة الحرف هى M = 4 صف المدخل التالى :

```
      INTEGER FORM(6)
      DATA FORM/'(1X,', 'I2,', '5X,', 'F6.', '2)', ' ')
```

تخزن المجموعة المتراصة FORM كما يلي :

FORM(1)	(	1	X	,
FORM(2)	I	2	,	
FORM(3)	5	X	,	
FORM(4)	F	6	.	
FORM(5)	2	)		
FORM(6)				

لاحظ أن FORM (6) تحتوى على مسافات (فراغات) فقط .

١٠ – ١٤ إفرض أن كلا من J و R مجموعة متراصة خطية بها أربعة عناصر وتخصص لها القـيم التالية :

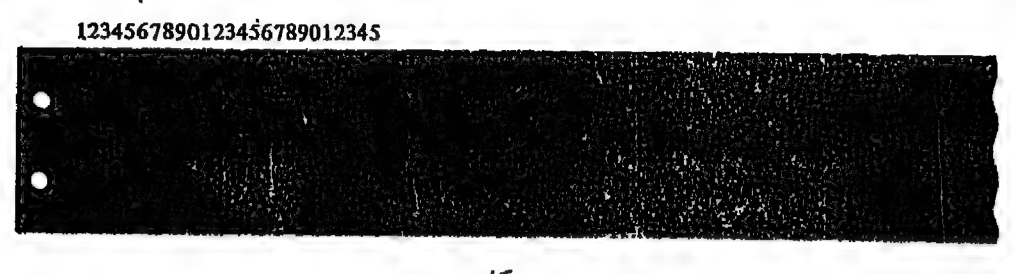
11, 2222, 77, 666 and 3333.33, 44.44, 5.555, 88888.888

على الترتيب . أوجد الخرج إذا نفذنا الآتى باستخدام المجموعة المتراصة FORM فى المسألة السابقة ١٠ – ١٣ .

```
      DATA K1/'I4,'/, K2/'F9.'/
      DO 100 I = 1, 4
              IF(J(I).GE.100) FORM(2) = K1
              IF(R(I).GE.1.0E3) FORM(4) = K2
              WRITE(6, FORM) J(I), R(I)
100   CONTINUE
```

سوف يمتد عرض الحقل من 2 إلى 4 إذا كانت J(I) ≥ 100 ويمتد عرض الحقل من 6 إلى 9 إذا كانت R(I) ≥ 1000 من ثم سيظهر الخرج كما فى الشكل ١٠ – ٤ . وفى كل الحالات ، تقرب قيمة R إلى مكانين عشريين .

1234567890123456789012345



شكل ١٠ – ٤

الرسم البيـانى

١٠ – ١٥ إفرض أننا نريد أن نستخدم حلقة DO لنحصل على قيم x من رقم صحيح J إلى رقم صحيح K بمعامل زيادة 1/N حيث N هى رقم صحيح موجب . أوجد (أ) عدد قيم x (ب) قيمة x بدلالة دليل INDEX حلقة DO (حـ) قيمة INDEX ، إن

وجدت، التي تتخطى محور y . اختبر النتائج في حالة $J = -2$ و $K = 8$ و $N = 4$ أي ، حيث تأخذ x القيم من -2 إلى 8 بمعامل زيادة قدره $1/4$ أي 0.25 :

(أ) هناك عدد $K - J$ وحدة مدى بين J و K وعدد N من قيم x تناظر كل من هذه الوحدات وهناك أيضاً قيمة إبتدائية لـ x وبذلك يكون هناك مجموع (إجمالي) لقيم x قدره :

$$N*(K - J) + 1$$

في المثال ، تأخذ x القيم :

$$-2, -1.75, -1.50, \ldots, 0, \ldots, 7.75, 8$$

هناك قيم إجمالية لـ x قدرها $4(8 + 2) + 1 = 41$

(ب) وفيما يلي العلاقة بين INDEX و x :

x:	J	J + 1/N	J + 2/N	J + 3/N	...	K
INDEX:	1	2	3	4	...	N*(K − J) + 1

لاحظ أن :

$$x = J + (INDEX - 1)/N$$

في المثـــال :

$$x = -2 + (INDEX - 1)/4$$

(جـ) يظهر محور y عند $x = 0$ ولكن x يمكن أن تساوى صفراً فقط إذا كانت $J \le 0 \le K$ ، في هذه الحالة ، هناك عدد $N*(-J)$ من قيم x تسبق $x = 0$ وبذلك ، يظهر محور y عندما يكون $INDEX = N*(-J) + 1$ يظهر محور y في هذا المثال عندما يكون $INDEX = 4(2) + 1 = 9$.

١٠ — ١٦ إفرض أن كل قيم y تقع بين الرقين الصحيحين L و M (حيث $L < M$) أوجد دالة مقياس الرسم التي تدرج كل قيمة لـ y إلى عدد صحيح J بين 1 و 101 . (نعتبرها أن عدد أعمدة الرسم البياني 101) اختبر دالة مقياس الرسم عند $L = -10$ و $M = 30$ أي ، حيث قيم y تقع بين -10 ، 30 .

بإضافة $-L$ إلى y نحصل على قيم بين 0 ، $M - L$. و بالضرب في $100/(M - L)$ نحصل على قيم بين 0 و 100. وبجمع 1 عن النتيجة نحصل على قيم بين 1 و 101 . وحيث أننا نريد أن نقرب y إلى عدد صحيح J يقع مابين 1 و 101 فضلاً عن البتر ، يجب أن نجمع 0.5 أيضاً . وبذلك تكون دالة مقياس الرسم كما يلي :

```
JSCALE(Y) = INT((Y + FLOAT(-L))*(100.0/FLOAT(M - L)) + 1.5)
```

في المثـــال :

```
JSCALE(Y) = INT((Y + 10.0)*(100.0/40.0) + 1.5)
```

١٠ — ١٧ اكتب البرنامج الذي يرسم القطع المكافىء بيانياً :

$$y = 3x^2 - x - 8$$

من $x = -3$ إلى $x = 3$ إفرض أن y تقع ما بين -10 ، 30 واحسب y لقيم x التي تتغير بمقدار 0.1 .

تستخدم حلقة DO (بالدليل I) لقيم x المختلفة . وفيما يلى العلاقة بين I و x :

x:	−3	−2.9	−2.8	...	0	...	2.9	3
I:	1	2	3	...	31	...	60	61

لاحظ أن هناك 61 قيمة لـ x ، يظهر محور y عند I = 31 ويمكن أن نحصل على x من I بواسطة :

x = −3.0 + FLOAT(I − 1)*0.1

(أنظر مسألة ١٠ − ١٥)

بفرض أن رسمنا البيانى سيحتوى على 101 عمود ، يمكن أن تدرج كل قيمة لـ y إلى عدد صحيح J يقع ما بين 1 و 101 باستخدام دالة مقياس الرسم التالية :

JSCALE(Y) = INT((Y + 10.0)*(100.0/40.0) + 1.5)

وكما سبق أن ناقشنا ذلك فى المسألة ١٠ − ١٦ . لاحظ أن JSCALE (0.0) = 26 لذا يكون العمود 26 من الرسم البيانى هو محور x .

وفيما يلى برنامج الفورتران الذى يرسم القطع المكافىء . الإجراء هو نفس الإجراء الذى تمت مناقشته فى قسم ١٠ − ٨ . وبالتالى : سيكون هناك سطر من النقط (.....) لكل محور ، وحرف x لكل نقطة من الرسم .

```
C       GRAPH OF A PARABOLA
        INTEGER DOT, CROSS, BLANK, LINE(101), LINEY(101)
        DATA DOT, CROSS, BLANK, LINE, LINEY/'.', 'X', ' ', 101*' ', 101*'.'/
C       BEGIN NEW PAGE
        WRITE(6, 10)
   10   FORMAT(1H1, 10X, 'GRAPH OF A PARABOLA'///3X, 'X', 6X, 'Y'//)
        DO 100 I = 1, 61
              X = -3.0 + FLOAT(I - 1)*0.1
              Y = 3.0*X*X - X - 8.0
              J  = INT((Y + 10.0)*(100.0/40.0) + 1.5)
              IF(I.EQ.31) GO TO 50
              LINE(26) = DOT
              LINE(J) = CROSS
              WRITE(6, 20) X, Y, LINE
   20         FORMAT(1X, 2(F4.2, 2X), 101A1)
              LINE(J) = BLANK
              GO TO 100
   50         LINEY(J) = CROSS
              WRITE(6, 20) X, Y, LINEY
  100   CONTINUE
        STOP
        END
```

١٠ − ١٨ اكتب البرنامج الذى يقرأ رقما صحيحاً N ويطبع عدداً قدره N^2 من النجوم بحيث يكون هناك عدد N من الأسطر لكل منها عدد N نجمة فى الأعمدة 1 و3 و5 و ... و 2N − 1 يجب أن تبدأ الصورة فى السطر الخامس من صفحة جديدة.

نحجز أولا مجموعة متراصة خطية LINE بها 132 عنصر (يناظر كل عنصر منها عمود على صفحة الطباعة) . وبعد ذلك تخزن مسافة فى كل عنصر من LINE . بعد قراءة N ، تخزن حينئذ نجمة فى العناصر LINE(1) و LINE(3) و ... و LINE (2N − 1) وفيما يلى البرنامج :

```
C       PROGRAM PRINTING A SQUARE OF ASTERISKS
        DIMENSION LINE(132)
        INTEGER AST
        DATA LINE/132*' '/, AST/'*'/
        READ(5, 10) N
   10   FORMAT(I5)
        NN = 2*N - 1
        DO 100 K = 1, NN, 2
              LINE(K) = AST
  100   CONTINUE
C       SKIP 5 LINES ON NEW PAGE
        WRITE(6, 20)
   20   FORMAT('1'/////)
C       PRINT THE ARRAY N TIMES USING A-FIELD
        DO 200 K = 1, N
              WRITE(6, 30) LINE
   30         FORMAT(1X, 132A1)
  200   CONTINUE
        STOP
        END
```

مســائل تكميلية

جمــل DATA

١٠ – ١٩ اكتب جملة DATA لإعطاء قيم إبتدائية كما يلي :

(أ) J = 5 و K = 5 و L = 5 و M = .TRUE. (حيث M متغير منطق)

(ب) A(I, J) = 0 لكل عنصر من المجموعة المتراصة A(4 × 5) .

١٠ – ٢٠ أوجد المدخلات فيما يلي :

(أ)
```
DATA L, M, N/2*2, '2*2'/, X, Y, Z/2.5, 2*3.5/
```
(ب)
```
DIMENSION X(6)
DATA X, Y, Z/5*2.22, 2*3.3, 4.4/
```
(جـ)
```
LOGICAL K(4)
DATA J, K, L/ 555, 3*.FALSE., .TRUE., 'TRUE'/
```

حقــل T—

١٠ – ٢١ إفرض أنه تم تثقيب بطاقة بيانات كالآتى :

```
                   1                   2                   3
 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 NOW  AND  THEN      4 4 4 4 4 5 5 5 5 5
```

أوجد المدخلات لو نفذنا الآتى :

(أ) (a)
```
        READ(5, 10) A, B
   10   FORMAT(T16, 2F5.2)
```

(ب) (b)
```
        READ(5, 20) X, J, K
   20   FORMAT(T14, F4.2, 2I3)
```

(جـ) (c)
```
        LOGICAL J, K
        READ(5, 30) I, J, K, C, D
   30   FORMAT(T16, I3, T4, 2L4, T16, 2F3.1)
```

١٠ – ٢٢ أوجد الخرج لكل جزء برنامج مما يلي :

(أ)

```
    J = 1111
    K = 5555
    L = 8888
    WRITE(5, 10) J, K, L
10  FORMAT(T16, I5, T6, I5, T26, I5)
```

(ب)

```
    A = 22.22
    B = 77.77
    C = 11.11
    WRITE(5, 20) A, B, C
20  FORMAT(T16, F5.1, 5X, F5.1, T1, F5.1)
```

حقــل —G :

١٠ – ٢٣ إفرض أنه تمَّ تثقيب بطاقة بيانات كالآتى :

```
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
1 1 1 1   4 4 4 4   6 6 6 E 2   T H A T
```

أوجد المدخل لو نفذنا الآتى :

(أ)

```
    LOGICAL J, K
    READ(5, 10) N, X, Y, J, K
10  FORMAT(5G5.2)
```

(ب)

```
    LOGICAL J, K
    READ(5, 20) X, N, Y, J, K
20  FORMAT(G7.1, 4G4.2)
```

١٠ – ٢٤ أوجد الخرج إذا طبعت قيم N و J و K و X و Y المسألة السابقة ١٠ – ٢٣ باستخدام مايلى :

```
    WRITE(6, 30) N, J, K, X, Y
30  FORMAT(1X, 5G10.3)
```

١٠ – ٢٥ إفرض مجموعة متراصة خطية A بها خمسة عناصر تحتوى بالترتيب على مايلى :

$44.444,\ 0.66666 \times 10^{-6},\ 33.3 \times 10^{2},\ 22.2 \times 10^{8},\ 333.3 \times 10^{-2}$

أوجد الخرج إذا كان الأمر هو مايلى :

```
    DO 100 K = 1, 5
        WRITE(6, 50) A(K)
50      FORMAT(11X, G12.4)
100 CONTINUE
```

معــامل التــدريج :

١٠ – ٢٦ أوجد القيمة الداخلية إذا أعطيت كود الشكل (مواصفات الحقل) والرقم الخارجى كمدخلات :

	(أ)	(ب)	(ﺟ)	(د)	(ﻫ)
كود الشكل :	2PF5.1	−3PF5.1	3PE5.1	−4PE10.1	−1PE10.1
المدخــل :	77.77	77.77	77.77	77.77E + 2	7777E + 2

١٠ – ٢٧ إفرض أن الرقم الداخل هو 444.888 أوجد الخرج إذا كانت مواصفات الحقل كل يلى :

(أ) 3PF10.2
(ب) −2PF10.1
(ﺟ) −PF10.3
(د) 4PE15.3
(ﻫ) −1PE15.5
(و) 2PE15.4

١٠ – ٢٨ إفرض أن X تحتوى 444.777 . أوجد الخرج إذا كان الأمر هو مايلى :

```
      WRITE(6, 10) X, X, X, X, X, X
10    FORMAT(1X, F12.2, 3X, E12.4, 3X, 2PF12.2, 3X, E12.4, 3X, 0PF12.2, 3X, E12.4)
```

قاعدة الأقـــواس اليسرى

١٠ – ٢٩ إدرس جزء البرنامج التالى :

```
DIMENSION K(6)
DATA K/4*33, 2*66/
WRITE(6, 10) K
```

أوجد الخرج إذا كانت جملة FORMAT المصاحبة هى :

(أ) `10 FORMAT(1X, I2, (I4, I6))`   (ب) `10 FORMAT(1X, I2, 2(3X, I2), 8X, I2)`

١٠ – ٣٠ إدرس جملة READ التالية :

```
READ(5, 20) A, B, C, X, Y, Z
```

واكتب جملة FORMAT بحيث :

تقرأ A من أول بطاقة باستخدام F10.2

وتقرأ B و C و X من ثانى بطاقة باستخدام F10.3 و F10.2 و F10.4

وتقرأ Y و Z من ثالث بطاقة باستخدام F10.3 ، F10.2

صيغ (FORMAT) وقت التنفيذ :

١٠ – ٣١ بفرض أن سعة الحرف 4 = M صف المدخل :

```
INTEGER FORM(5)
DATA FORM/'(1X,', 'I3,', '3X,', 'F6', '.3)'/
```

١٠ – ٣٢ إفرض أن J و X مجموعات متراصة خطية كل منها بها خمسة عناصر ، وأفرض أن J و X قد أعطيت القيم التالية :

333, 33333, 33, 333333, 3   and   4.4, 66666.66, 1.1, 88.88888, 2222.2

على الترتيب . أوجد الخرج إذا نفذنا الآتى باستخدام المجموعة المتراصة FORM فى المسألة السابقة ١٠ – ٣١ :

```
      DATA IA, IB/'I7,', 'F10'/
      DO 100 K = 1, 5
         IF(J(K).GE.1000) FORM(2) = IA
         IF(X(K).GE.100.0) FORM(4) = IB
         WRITE(6, FORM) J(K), X(K)
100   CONTINUE
```

١٠ – ٣٣ إفرض أن مجموعة متراصة خطية J تحتوى 75 رقماً صحيحاً موجباً أقل من 10.000 . أكتب جزء البرنامج الذى يطبع الأرقام فى عمود بحيث يكون مضبطا من اليسار ، كما يلى مثلا :

28
111
5
3344
62
⋮

الرسم البيـانى

١٠ – ٣٤ اكتب البرنامج الذى يرسم الدالة $y = \text{Sin } x$ بالنسبة إلى $-10 \le x \le 10$ (تذكر أن $-1 \le y \le 1$ لأى قيمة لـ x)

١٠ – ٣٥ إفرض أن القيم y_1 و y_2 و . . . و y_N خزنت بمجموعة متراصة Y بها N عنصر . أكتب البرنامج الفرعى SUBROUTINE

ENDS(Y, N, YMIN, YMAX)

الذى يجد القيمة الصغرى YMIN والقيمة العظمى YMAX للعناصر فى المجموعة المتراصة Y .

١٠ – ٣٦ اكتب البرنامج الذى يرسم الدالة :

$$y = x^3 + 2x^2 - 15x - 8$$

من $x = -5$ إلى $x = 4$ حيث تتغير قيم x بمقدار 0.2 . بالتحديد خزن كل قيم y أولا فى مجموعة متراصة خطية وبعد ذلك استخدم البرنامج الفرعى ENDS فى المسألة ١٠ – ٣٥ ودالة مقياس الرسم المشابهة إلى الدالة المشروحة فى المسألة ١٠ – ١٦ لتدريج y إلى رقم صحيح J يتراوح بين 1 و 101 .

١٠ – ٣٧ اكتب البرنامج الذى يقبل رقمين صحيحين موجبين J و K ويطبع مستطيل J*K من النجوم بحيث أن تكون هناك عدد J من الأسطر بعدد K من النجوم تظهر فى الأعمدة 1 و 3 و 5 و .... و 2K – 1 (يجب أن نبدأ الصورة فى السطر الثالث من صفحة جديدة) .

اجابات المسائل التكميلية المختارة

١٠ – ١٩

(أ) DATA J, K, L, M/3*5, .TRUE./ (ب) DATA A/20*0.0/

١٠ – ٢٠ (أ) تحتوى L و M على 2 وتحتوى N على الحروف الأربعة $2*2_b$ باعتبار أن السمة M = 4) ، وتحتوى X على 2.5 وتحتوى Y و Z على 3.5) .

(ب) تحتوى X(1) و . . . X(5) على 2.22 وتحتوى X(6) و Y على 3.3 وتحتوى Z على 4.4 .

(جـ) تحتوى J على 555 وتحتوى K(1) و K(2) و K(3) على .FALSE. وتحتوى K(4) على .TRUE. وتحتوى L على سلسلة الحروف TRUE « باعتبار أن السمة M = 4 » .

١٠ – ٢١ (أ) تحتوى A على 444.44 وتحتوى B على 555.55 .

(ب) تحتوى X على 0.44 وتحتوى J على 444 وتحتوى K على 555 .

(ﺟ) تحتوى I على 444 وتحتوى J على .FALSE. وتحتوى K على .TRUE. وتحتوى C على 44.4 وتحتوى D على 44.5 .

١٠ – ٢٢

(أ) $_{bbbbbb}5555_{bbbbbb}1111_{bbbbbb}8388$ (ب) $11.1_{bbbbbbbbbbb}22.2_{bbbbbb}77.8$

١٠ – ٢٣

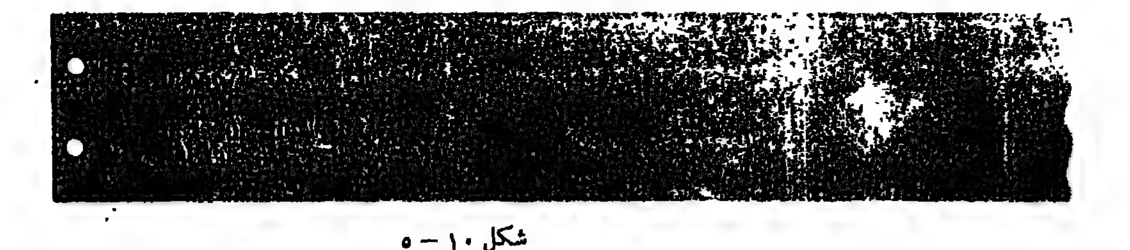
(أ) 11110, 444.40, 6.66E2(or: 666.0), .TRUE., .FALSE.

(ب) 111104.4, 4406, 0.66E2(or: 66.0), .TRUE., .TRUE.

١٠ – ٢٤ مضبطتين من اليمين فى الحقول الخاصة بها .

(أ) $11110, T, F, 444._{bbbb}, 666._{bbbb}$ (ب) $4406, T, T, 0.111E_{b}06, 66.0_{bbbb}$

١٠ – ٢٥ أنظر شكل ١٠ – ٥



شكل ١٠ – ٥

١٠ – ٢٦

0.7777; 77770.: 0.07777; 77.77E + 2, i.e., 7777.0 (no effect); 777.7E + 2, i.e., 77770.0 (no effect)

١٠ – ٢٧ (أ) 444888.00 (ﺟ) 44.489 (ﻫ) 0.044489E 04

(ب) 4.4 (د) 4450.E – 01 (و) 44.49E 01

١٠ – ٢٨ $_{bbbbbb}444.78_{bbbbb}0.4448E_{b}03_{bbbbbbb}44477.70_{bbbbbb}44.48E_{b}01_{bbbbbbbbb}444.78_{bbbbb}0.4448E_{b}03$

١٠ – ٢٩ (أ) $33_{bb}33_{bbbb}33$ على سطر ، $_{b}33_{bbbb}66$ على السطر التالى و $_{b}66$ على السطر الثالث .

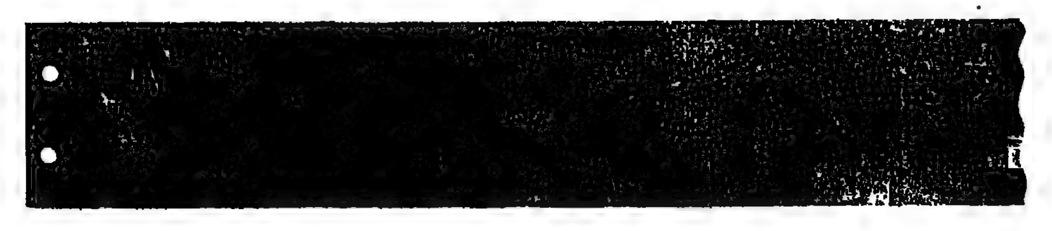
(ب) $33_{bbb}33_{bbb}33_{bbbbbbb}33$ على سطر و $_{bb}66_{bbb}66$ على السطر التالى .

١٠ – ٣٠ 20 FORMAT(F10.2/(F10.3, F10.2, F10.4/))

١٠ – ٣١ يخزن المجموعة المراصة FORM كما يلي :

FORM(1)	(	1	X	,
FORM(2)	I	3	,	
FORM(3)	3	X	,	
FORM(4)	F	6		
FORM(5)	.	3	)	

١٠ – ٣٢ أنظر شكل ١٠ - ٦



شكل ١٠ – ٦

١٠ – ٣٢ تلميح : استخدم صيغ (FORMAT) وقت التنفيذ مع كود الشكل I_1 و I_2 و I_3 و I_4 تبعاً لقيمة J التي تحتوى واحد أو إثنين أو ثلاثة أو أربعة أرقام على الترتيب .

الفصل الحادى عشر

ملامح متنوعـــة للفورتران

١١ ــ ١ مقدمــة

يتناول هذا الفصل خصائص إضافية متعددة للغة البرمجة فورتران . وبالتحديد ، سوف ندرس بصورة عامة جمل النوع وجملة IMPLICIT ومتغيرات متضاعفة الدقة DOUBLE PRECISION ومتغيرات مركبة COMPLEX وجمل المتغيرات المشتركة COMMON والمتكافئة EQUIVALENCE .

١١ ــ ٢ جمل النوع (TYPE)

هناك أنواع متعددة من المتغيرات حقيقية REAL وصحيحة INTEGER ومنطقية LOGICAL ومتضاعفة الدقة DOUBLE PRECISION ومركبة COMPLEX . سبق وناقشنا المتغيرات الحقيقية REAL والصحيحة INTEGER فى الفصل الثانى ، وتمت مناقشة المتغيرات المنطقية LOGICAL فى الفصل التاسع ، وسوف نناقش فى هذا الفصل المتغيرات المركبة ومتضاعفةالدقة COMPLEX و DOUBLE REIECSION فى الأقسام ١١ – ٤ و ١١ – ٥ .

يسمح لنا الفورتران باستخدام « الحرف الأول » كتقليد يدل على نوع المتغير وبالتحديد ، فأى متغير يبدأ اسمه بأى حرف من الحروف I أو J أو K أو L أو M أو N فهو متغير صحيح أما إذا ابتدأ بأى حرف آخر فهو متغير حقيقى . ومع ذلك ، فيمكن التغلب على هذا التقليد باستخدام جملة النوع (Type) .

ويكون لجملة النوع (Type) الشكل التالى :

TYPE قائمة متغيرات

حيث تفصل أسماء المتغيرات بفصلات . وتعلن جملة النوع عن نوعية المتغيرات فى القائمة فمثلا :

```
REAL MONEY, RATE, NEW
INTEGER X, Y, NEXT
LOGICAL A, B
DOUBLE PRECISION S, T
COMPLEX ROOT, COEF
```

تبين أن MONEY و RATE و NEW متغيرات حقيقية ، وأن X و Y و NEXT متغيرات صحيحة ، وهكذا . لاحظ أنه ليس من الضرورى تضمين RATE فى الجملة الأولى أو NEXT فى الجملة الثانية ، يشعر بعض المبرمجين فى الحقيقة أنها ممارسة جيدة أن تذكر كل المتغيرات فى جملة نوع بغض النظر عن حرفه الأول .

يمكن أن تظهر أيضاً أسماء المجموعات المتراصة فى جمل النوع (type) . فى الحقيقة ، يمكن أن تتجنب استخدام جملة الأبعاد DIMENSION منفصلة بتضمين حجم المجموعة المتراصة فى جملة النوع فمثلا :

```
INTEGER COUNT(30)
REAL NUMBER(4, 5), LAST
```

تعلن أن COUNT مجموعة متراصة خطية بها 30 عنصراً وأن NUMBER مجموعة متراصة حقيقية (4 × 5) .
جمل النوع (type) جمل غير منفذة ولذا يجب أن تظهر فى بداية البرنامج قبل جملة DATA وقبل أى استخدام للمتغيرات .

١١ — ٣ جملة IMPLICIT

يمكن أن نتوسع فى استخدام التقليد للنوع أى « الحرف الأول » باستخدام جملة IMPLICIT . وسوف نوضح ذلك بمثال . ادرس جملة IMPLICIT التالية :

```
IMPLICIT INTEGER(B, D, W – Z), LOGICAL(A, N – P), COMPLEX(F – H)
```

هذه الجملة تخبر المترجم أن كل أسماء المتغيرات التى تبدأ بالحروف B و D و W و X و Y و Z متغيرات صحيحة ، وأسماء المتغيرات التى تبدا بالحروف A و N و O و P هى متغيرات منطقية ، وأسماء المتغيرات التى تبدأ بالحروف F و G و H هى متغيرات مركبة إلا إذا نص على غير ذلك صراحة فى جمل نوع . لاحظ أن مدى الحروف يحدد بكتابة أول وآخر حرف منفصلين بإشارة ناقص . لاحظ أيضاً أن الحروف ومداها يتبع النوع TYPE ومفصولة بفصلات ومحاطة بين أقواس .

ويجب أن نذكر الآتى عند استخدام جملة IMPLICIT :

١ — عادة يمكن أن تظهر جملة IMPLICIT واحدة فى أى برنامج أو فى أى برنامج فرعى . ويجب أن تكون أول جملة ممكنة ، بمعنى أنها يجب أن تكون أول جملة فى البرنامج الرئيسى . ويجب أن تأتى فى البرنامج الفرعى بعد جملة تعريف البرنامج الفرعى مباشرة .

٢ — يمكن أن تتغلب على جملة IMPLICIT بجملة نوع . علاوة على ذلك ، يبقى تقليد النوع العادى أى « الحرف الأول » لأى متغيرات لم تظهر فى جملة IMPLICIT أو فى جمل نوع . كمثال . افرض برنامجاً به الجمل التالية :

```
IMPLICIT DOUBLE PRECISION (A – D, S), INTEGER(F – G)
COMPLEX ALPHA, BETA
INTEGER SET, CLASS
```

إذن ALPHA و BETA متغيرات مركبة و SET و CLASS متغيرات صحيحة فى ضوء جمل النوع . ANSWER و SUM متغيرات متضاعفة الدقة DOUBLE PRECISION فى ضوء جملة IMPLICIT . من ناحية أخرى RATE و KOUNT لم يتضمنا فى أى من الجمل ، ولذا فهما متغيرات حقيقية REAL وصحيحة INTEGER على الترتيب .

٣ — معاملات البرنامج الفرعى وأيضاً اسم البرنامج الفرعى FUNCTION يتم تغطيتها فى جملة IMPLICIT فى البرنامج الفرعى .

١١ — ٤ الدقة المتضاعفة DOUBLE PRECISION

يمكن أن يخزن الحاسب عادة سبعة أو ثمانية أرقام معنوية فى اى خبية ذاكرة وسوف تحتفظ حساباته (وتسمى أيضاً حسابات الدقة المفردة) برقم مماثل من الأرقام المعنوية . ومع ذلك ، يسمح لنا الفورتران بالحصول على نتائج أكثر دقة باستخدام ما يسمى قيم وحسابات الدقة المتضاعفة . فى هذه الحالة تخزن كل قيمة فى مكانين (2) من أماكن الذاكرة ، وعلى ذلك يضاف تقريباً عدد الأرقام المعنوية . (فمثلا ، تحتفظ سلسلة IBM360/370 بسبعة أرقام معنوية تحت الحساب العادى ولكن تحتفظ بعدد 16 رقماً معنوياً تحت حساب الدقة المتضاعفة) . سنناقش فى هذا القسم ، القوانين الخاصة بحساب الدقة المتضاعفة .

ملاحظة : حيث أن قيم الدقة المتضاعفة تحتل عدداً مضاعفاً من اماكن الذاكرة وحيث أن الحساب يأخذ ، وقتاً أطول فى التنفيذ ، يتراوح هذا الوقت ما بين مرتين إلى عشر مرات ، فذلك يلزمنا أن نستخدم الدقة المتضاعفة فى الحالة التى تتطلب هذه الدقة فقط .

متغيرات

نعلن عن المتغيرات والمجموعات المتراصة الخاصة بالدقة المتضاعفة بواسطة جملة DOUBLE PRECISION كما ناقشنا فى قسم ١١ — ٢ .

ثوابت

تكتب ثوابت الدقة المتضاعفة بطرق مشابهة للثوابت الحقيقية مفردة الدقة ولكن يجب أن توجد علامة عشرية فى أى ثابت دقة متضاعفة .

١ — بدون اس . يفسر الثابت الحقيقى المكتوب بدون أس كثابت بدقة مفردة ، إلا إذا تجاوز عدد أرقامه المعنوية العدد الأقصى المسموح به لثابت حقيقى بدقة مفردة . فمثلا بالنسبة لسلسلة IBM360/370 التى تسمح بسبعة أرقام معنوية ، سيعتبر الرقم 123.465 ثابت بدقة مفردة بينما سيعتبر الرقم 123.456000000 ثابت بدقة متضاعفة .

٢ — ذات أس . تمثل ثوابت الدقة المفردة فى الشكل الأسى باستخدام الحرف E فمثلا 46.8086E7 ثابت بدقة مفردة . وتمثل ثوابت الدقة المتضاعفة فى الشكل الأسى باستخدام الحرف D بدلا من الحرف E وبذلك ، تمثل 46.8086D7 نفس الرقم بالدقة المتضاعفة ، ومع ذلك ، فإنه يخزن داخلياً بعدد 16 رقماً معنوياً بالكامل . كما يلى :

$$0.4680860000000000 \times 10^9$$

(أى تضا[illegible] أصفار لحصول على 16 رقما صحيحاً تماماً كما هو المطلوب) .

يجب أن نتذكر أنه لا توجد أرقام صحيحة بدقة متضاعفة .

إدخال / إخ[illegible]

يستخدم حـ[illegible] D لإدخال / وإخراج قيم الدقة المتضاعفة بعيداً عن حقيقة أن حقول D تسمح بتناول أماكن عشرية أطول ، إلا أن استخدام حقل D مشابه تماماً استخدام حقل E فمثلا ، مواصفات الحقل لها الشكل .

Dw.d

حيث *w* هى عرض الحقل و *d* هى عدد الأرقام العشرية ولذلك ننصح القراء بمراجعة الفصل الثالث الخاص بتوصيف حقول — E لمناقشة هادفة .

عمليات حسابية

تنفذ حسابات الدقة المتضاعفة عندما يكون طرفاً المعادلة من نوع الدقة المتضاعفة ، وتسمح معظم المترجمات بمزيج من حدود الدقة المفردة والدقة المتضاعفة . فى مثل هذه الحالة ، يحول حد الدقة المفردة أولا إلى دقة متضاعفة . (بإضافة أصفار خلف الرقم) وبعد ذلك تجرى حسابات الدقة المتضاعفة . من ناحية أخرى ، لا تسمح معظم المترجمات بمزيج من الأنواع الصحيحة والدقة المتضاعفة ، فيما عدا أنه يمكن أن نرفع قيمة بدقة متضاعفة إلى أس صحيح . نذكر هنا أيضاً أن أى قيمة سالبة بدقة متضاعفة لا يمكن أن ترفع إلى قوة حقيقية أو قوة بدقة متضاعفة كما هو الحال مع الأرقام السالبة الحقيقية (انظر صفحة ٢٨) .

جمل تخصيص

وكما فى حالة الدقة المفردة ، فيمكن للانواع على جانبى إشارة = أن تكون مختلفة . وتحول القيمة التى سوف تخزن دائماً إلى نوع مكان التخزين المناظر . بالإضافة إلى ذلك فإن قيم الدقة المتضاعفة تبتر ولا تقرب ، عند تحويلها إلى دقة مفردة . ادرس مثلا الجزء التالى :

```
DOUBLE PRECISION A
A = 1.0
B = 123.456789098
J  = 123.456789098
K = 3.4444D3
```

وعلى ذلك

١ – يحول الثابت (1.0) إلى دقة متضاعفة ويخزن فى A .

٢ – يخزن الرقم 123.4567 فقط فى B أى ، يبتر الثابت بعد سبعة أرقام معنوية .

٣ – يخزن الرقم 123 فقط فى J أى ، يبتر الثابت عند العلامة العشرية .

٤ – رغم أن 3.4444D3 تساوى 3444.4 ولكن تخصص 3 فقط إلى K حيث يبتر الثابت عند العلامة العشرية .

الدوال المكتبية

تحول الدوال المكتبية DFLOAT و DBLE القيم الصحيحة وقيم الدقة المفردة إلى قيم دقة متضاعفة على الترتيب وتحول الدالة SNGL قيمة بدقة متضاعفة إلى دقة مفردة . هناك أيضاً دوال مكتبية تسمح دائماً بإجراء حسابات الدقة المتضاعفة لمعظم الدوال المستخدمة . فمثلا ، إذا كانت DX خلاصة دالة بدقة متضاعفة فسوف تعطى :

```
DSQRT(DX)
```

الجذر التربيعى لـ DX باستخدام حساب الدقة المتضاعفة . بالمثل تعطى DABS(DX) و DLOG(DX) و DSIN(DX) القيمة المطلقة واللوغاريتم ، وجيب الزاوية لمتغير DX بدقة متضاعفة . وبصورة عامة فالحرف D إذا وضع أمام أى دالة فهو يدل على حسابات الدقة المتضاعفة . وسوف نعطى قائمة بدوال أخرى كثيرة فى الملحق أ وسوف نضطر إلى استخدام :

```
DLOG(DBLE(X))
```

للحصول على اللوغاريتم الطبيعى بالدقة المتضاعفة لقيمة صحيحة بدقة مفردة X حيث أن خلاصة الدالة DLOG يجب أن تكون بدقة متضاعفة .

برامج فرعية

يمكن أن تنقل قيم الدقة المتضاعفة من وإلى برنامج فرعى عن طريق الخلاصات (تماماً مثل أى أنواع أخرى من القيم) ، ومع ذلك ، فيجب أن نعلن عن أسماء هذه الخلاصات وأسماء المعاملات المناظرة لها كمتغيرات بدقة متضاعفة فى كل من البرنامج الرئيسى والبرنامج الفرعى. يمكن أن نعلن أيضاً أن البرنامج الفرعى FUNCTION نفسه ذو دقة متضاعفة بكتابة DOUBLE PRECISION فى بداية جملة تعريف FUNCTION أو بإعلان أن اسم البرنامج NAME ذو دقة متضاعفة بداخل البرنامج الفرعى . فمثلا فيما يلى ، برنامجان فرعيان متكافئان .

```
DOUBLE PRECISION FUNCTION SUM(A, B)
DOUBLE PRECISION A, B
SUM = A + B
RETURN
END
```

```
FUNCTION SUM(A, B)
DOUBLE PRECISION SUM, A, B
SUM = A + B
RETURN
END
```

وعلاوة على ذلك ، تتطلب معظم المترجمات من البرنامج الداعى أن يعلن عن اسم (NAME) البرنامج الفرعى (الدالة) ذى الدقة المتضاعفة FUNCTION . أى يجب أن تظهر الجملة :

```
DOUBLE PRECISION SUM
```

أو ما يكافئها فى البرنامج الداعى الذى يستخدم البرنامج الفرعى المذكور أعلاه SUM (وليس هذا المطلب عاماً، بل يجب على المبرمج أن يبحث عن المتطلبات الدقيقة للمترجم الخاص به) . من ناحية أخرى ، لا يوجد برنامج فرعى بدقة متضاعفة DOUBLE PRECISION SUBROUTINE حيث لا تخصص قيمة لاسم (NAME) البرنامج الفرعى .

مثال ١١ - ١

بفرض أننا استخدمنا سلسلة IBM 360/370 فسوف نحصل من جزء البرنامج التالى :

```
    DOUBLE PRECISION DA, DB
    DA = 2./3.
    DB = 2.0D0/3.0D0
    WRITE(6, 10) DA, DA
    WRITE(6, 10) DB, DB
10  FORMAT(1X, D23.16, 10X, D12.4)
```

على :

```
0.6666666000000000D 00        0.6667D 00
0.6666666666666666D 00        0.6667D 00
```

وهذا يحدث لأن 2./3. تحسب بالدقة المفردة وتبتر بعد سبعة أرقام معنوية أى يكون لها القيمة 0.6666666 أما عندما تخزن 2./3. فى DA فتحول إلى دقة متضاعفة بإضافة أصفار . من جهة أخرى تحسب 2.0D0/3.0D0 بالدقة المتضاعفة وتبتر بعد 16 رقماً معنوياً ، أى تكون لها القيمة 0.6666666666666

(ليس هناك تقريب فى هذه الحسابات) ومع ذلك فعندما طبعت هذه القيم تم تقريبها إلى أربعة أرقام معنوية كما وصفت فى الصيغة (FORMAT)

١١ ـ ٥ الاعداد المركبة

تسمح مترجمات الفورتران للعمليات الحسابية بتناول أعداد مركبة . دعنا نراجع أولا حساب الأعداد المركبة وبعد ذلك نناقش طريقة تناول الفورتران لها .

يكتب عادة عدد مركب z فى الشكل :

$$z = a + bi$$

حيث a و b أعداد حقيقية عادية و $i = \sqrt{-1}$. تسمى a الجزء الحقيقى لـ z وتسمى b الجزء التخيلى لـ z (يمكن أن نعرف الجزء الحقيقى a فى صورة العدد المركب $(a + 0i)$)

يمكن الحصول على جمع وضرب أعداد مركبة باستخدام القوانين المجمعة والموزعة العادية للجبر مع اعتبار $i^2 = -1$:

$$(a + bi) + (c + di) = a + c + bi + di = (a + c) + (b + d)i$$
$$(a + bi)(c + di) = ac + bci + adi + bdi^2 = (ac - bd) + (bc + ad)i$$

مرافق العدد المركب $z - a + bi$ يرمز إليه ويعرف بالآتى :

$$\bar{z} = a - bi$$

(لاحظ أن $z\bar{z} = a^2 + b^2$. إذا كانت $z \neq 0$ أى ، إذا كانت $a \neq 0$ أو $b \neq 0$ فإنه يمكن إيجاد مقلوب z أى z^{-1}) والقسمة على z مبينة فيما يلى :

$$z^{-1} = \frac{\bar{z}}{z\bar{z}} = \frac{a}{a^2 + b^2} + \frac{-b}{a^2 + b^2}i \quad \text{،} \quad \text{و} \quad \frac{w}{z} = wz^{-1}$$

حيث w عدد مركب أيضاً . ونعرف أيضاً :

$$-z = -1z \quad \text{و} \quad w - z = w + (-z)$$

قيمة z المطلقة (أو القيمة المطلقة لـ z حيث $(z = a + bi)$ تكتب بالصورة $|z|$ وتعرف بواسطة :

$$|z| = \sqrt{a^2 + b^2} = \sqrt{z\bar{z}}$$

مثال ١١ ـ ٢

افترض أن $z = 5 - 3i$ و $w = 5 - 2i$ من ثم يمكن استنتاج التالى :

$$z + w = (2 + 3i) + (5 - 2i) = 2 + 5 + 3i - 2i = 7 + i$$
$$zw = (2 + 3i)(5 - 2i) = 10 + 15i - 4i - 6i^2 = 16 + 11i$$
$$\bar{z} = \overline{2 + 3i} = 2 - 3i \quad \text{and} \quad \bar{w} = \overline{5 - 2i} = 5 + 2i$$
$$\frac{w}{z} = \frac{5 - 2i}{2 + 3i} = \frac{(5 - 2i)(2 - 3i)}{(2 + 3i)(2 - 3i)} = \frac{4 - 19i}{13} = \frac{4}{13} - \frac{19}{13}i$$
$$|z| = \sqrt{4 + 9} = \sqrt{13} \quad \text{and} \quad |w| = \sqrt{25 + 4} = \sqrt{29}$$

حيث أن العدد المركب يتكون من جزءين ، جزء حقيق وجزء تخيل ، فإننا نحتاج إلى مكانين فى الذاكرة لتخزين العدد المركب . وفيما يلى القواعد المتعلقة بالثوابت والمتغيرات المركبة وطريقة حسابها .

متغيرات

يعلن عن المتغيرات والمجموعات المتراصة المركبة باستخدام جملة COMPLEX كما تمت مناقشتها فى قسم ١١ — ٢ .

ثوابت

يمثل الفورتران الثابت المركب بزوج مرتب من الثوابت الحقيقية تفصل بينهما فصلة ويحاطا بأقواس . يشير الرقم الحقيقى الأول إلى الجزء الحقيقى من العدد المركب ويشير الرقم حقيقى الثانى إلى الجزء التخيل . يمكن أن يكتب كل من الرقين بأس أو بدون أس ولكن لا يمكن أن يكون رقماً صحيحاً . وبذلك يمكن كتابته :

$1 + 2i$

فى الفورتران بأى من الطرق التالية :

```
(1.0, 2.0)
(0.1E1, 2.0)
(1.0, 2.0E0)
(0.1E1, 0.2E1)
```

ولكن لا يمكن تمثيلها كما يلى :

```
(1, 2)
```

من ناحية أخرى لا يقبل زوج مرتب من المتغيرات الحقيقية أو التعبيرات عند تعريف قيمة مركبة . فثلا إذا عرف أن X متغير حقيق فلا يمكن أن نستخدم :

```
(X, 2.0)
```

لتعريف قيمة مركبة . مع ذلك نجد فى الفورتران دالة مكتبية مركبة CMPLX تحول زوجاً مرتباً من تعبيرات حقيقية إلى ثابت له قيمة مركبة . بذلك تكون :

```
CMPLX(X, 2.0)   and   CMPLX(X + 2.0, X − 2.0)
```

طرق الفورتران لكتابة الأعداد المركبه $x + 2i$ و $(x + 2) + (x - 2)i$.

الإدخال / الإخراج

ليس هناك حقل معين لإدخال / إخراج الأعداد المركبة . ومع ذلك حيث أن كل رقين حقيقين يناظران عدداً مركباً ، فيجب أن نعطى مواصفات حقلين لكل متغير مركب . فثلا :

```
      COMPLEX Z
      READ(5, 10) Z
10    FORMAT(F10.2, 10X, F10.2)
```

سوف تخصص القيمة فى الأعمدة من 1 إلى 10 كجزء حقيق لـ Z والقيمة فى الأعمدة من 21 إلى 30 كجزء تخيل لـ Z . وبذلك إذا أعدت بطاقة البيانات كما يلى :

```
           1         2         3
123456789012345678901234567890123456789012345
        12.34     333.33     13.579
```

فسوف تكون قيمة Z هى القيمة $12.34 + 13.579i$. ومن الطبيعى أن تكون مواصفات الحقول إما حقل E أو حقل F أو حقل G أو توافقية منها .

العمليات الحسابية

يجرى الحساب المركب عندما يكون كل من طرفى العملية مركب ، أو عندما يكون أحدهما حقيقياً والآخر مركباً . فى الحالة الأخيرة ، تحول القيمة الحقيقية ولتكن a إلى مكافئها المركب $(a, 0.0)$ وبعد ذلك يتم تأدية الحساب المركب . فمثلا إذا كانت X حقيقية و C مركبة فتعبيرات الفورتران التالية مقبولة :

X + (1.0, 2.0),   X + C,   C − X,   X/C,   C*X,   C + 1.0

إذا كان المترجم يقبل مزيجاً من القيم الصحيحة والحقيقية يمكن أيضاً استخدام القيم الصحيحة مع القيم المركبة ، فمثلا تكون C + 1 مقبولة ويتم تأدية الحساب المركب . ومع ذلك هناك قيد واحد خاص يسرى على الأعداد المركبة ولا يسرى على الأرقام الحقيقية وهو : لا يمكن رفع رقم مركب إلى قوة حقيقية . بمعنى آخر فإن العدد المركب يرفع إلى قوة صحيحة فقط .

جمل التخصيص

يمكن أن تخصص قيم حقيقية أو صحيحة أو مركبة إلى متغيرات مركبة . فمثلا إذا كانت A و B متغيرات مركبة و X و Y متغيرات حقيقية ، فجمل التخصيص الآتية صحيحة :

```
A = (1.0, 0.2E1)
A = A + B**2
A = 25
A = 3*X
A = 3*B + Y − 8.0
A = CMPLX(X, Y + 2.0)
```

وبالتحديد ، تحول القيم الصحيحة أو الحقيقية إلى قيم مركبة قبل تخزينها فى المتغير المركب A . من ناحية أخرى ، لا يمكن تخصيص قيم مركبة إلى متغيرات حقيقية أو صحيحة وبذلك تكون الجمل التالية :

```
X = A + B
K = B**2
```

غير صحيحة .

الدوال المكتبية المركبة

سبق وأن ناقشنا الدالة CMPLX (X, Y) والتي تأخذ خلاصات حقيقية X و Y وتحولهما إلى رقم مركب X + Y*i* ويمكن أن نحصل على الجزء الحقيقي والتخيلي والقيمة المطلقة المتغير المركب بواسطة الدوال REAL ، AIMAG و CABS على الترتيب . أى إذا كانت C = X + Y*i* فسوف يحدث التالى :

REAL (C) سوف تعطى القيمة X

AIMAG (C) سوف تعطى القيـ

CABS (C) سوف تعطى القيمة $\sqrt{X^2 + Y^2}$

هناك دوال مكتبية مركبة أخرى مشابهة للدوال الحقيقية . فمثلا إذا كانت C متغيرا أو تعبيرا مركباً فسوف تعطى :

CSQRT(C)

الجذر التربيعى لدمية المركبة C . بالمثل CSIN (C) و CCOS (C) و CLOG (C) و CEXP (C) تحسب القيم المركبة لجيب ، وجيب تمام ، واللوغاريتم ، والأس ، الكمية C على الترتيب . وبصورة عامة فالحرف C أمام أى دالة يدل على أنها دالة مركبة . ويظهر فى الملحق أ قائمة بمثل هذه الدوال الكثيرة .

برامج فرعية

تستخدم القيم المركبة فى البرامج الفرعية بنفس طريقة استخدام قيم الدقة المتضاعفة . بالتحديد ، تنقل القيم المركبة من وإلى البرنامج الفرعى عن طريق الخلاصات ويجب أن تعرف أسماء هذه الخلاصات والمعاملات المناظرة بأنها مركبة فى كل من البرنامج الداعى والبرنامج الفرعى . وأيضاً يمكن أن يعرف برنامج فرعى FUNCTION بأنه مركب بكتابة COMPLEX فى بداية جملة تعريف FUNCTION أو بتعريف الاسم NAME على أنه مركب فى البرنامج الفرعى . فمثلا الجزءان التاليان متكافئان :

```
COMPLEX FUNCTION XYZ(A, B, C)     or     FUNCTION XYZ(A, B, C)
COMPLEX A, B                             COMPLEX XYZ, A, B
```

كل منها يعرف XYZ دالة ذات قيمة مركبة وأن A و B متغيرات مركبة . وعلاوة على ذلك فإن معظم المترجمات تتطلب أن يعرف اسم NAME البرنامج الفرعى المركب FUNCTION على أنه مركب فى البرنامج الداعى كما فى دوال الدقة المتضاعفة . أى أن الجملة :

```
COMPLEX XYZ
```

يجب أن تظهر فى البرنامج الداعى الذى يستخدم البرنامج الفرعى المركب XYZ المذكور أعلاه .

١١ — ٦ جملة GO TO المخصصة وجملة ASSIGN

تمدنا بعض المترجمات بجملة أخرى لنقل التحكم تسمى جملة GO TO المخصصة . جملة GO TO المخصصة وجملة ASSIGN المناظرة لها الشكل التالى :

ASSIGN n TO J

⋮

GO TO J, $(n_1, n_2, \ldots, n_k)$

حيث n و n_1 و n_2 و ... و n_k ثوابت صحيحة بدون إشارة تدل على أرقام جمل (عناوين) و J متغير صحيح . تخصص جملة ASSIGN الثابت n إلى المتغير J . عندما تنفذ جملة GO TO المخصصة ، ينقل التحكم إلى الجملة المرقمة بالرقم n أى القيمة الحالية لـ J .

تلخص قوانين جملة ASSIGN وجملة GO TO المخصصة كالآتى :

١ – يجب أن يكون المتغير J متغيراً صحيحاً (بدون دليل) .

٢ – لا يمكن تغيير المتغير J إلا بجملة ASSIGN أخرى . فمثلا الجزء التالى :

```
ASSIGN 25 to LAST
LAST = LAST + 5
GO TO LAST, (20, 25, 30, 35)
```

غير مسموح به

٣ – يجب أن تلى قيمة J عند ظهورها فى قائمة أرقام الجمل المتغير J فى جملة GO TO .

١١ – ٧ المداخل والرجوع المتعددة لبرنامج فرعى

(أ) مداخل متعددة

عادة ، عندما يظهر اسم برنامج فرعى (SUBROUTINE أو FUNCTION) فى برنامج داع فالمدخل إلى البرنامج الفرعى هو عند أول جملة منفذة تلى جملة تعريف البرنامج الفرعى . توفر بعض المترجمات الأوامر لدخول برنامج فرعى عند نقاط دخول مختلفة . وتستخدم جملة ENTRY فى البرنامج الفرعى للإشارة إلى نقاط الدخول البديلة . ولهذه الجملة الشكل التالى :

ENTRY NAME (قائمة معاملات)

حيث يختلف اسم المدخل (NAME) عن اسم البرنامج الفرعى . جملة ENTRY جملة غير منفذة لذلك توضع فى البرنامج الفرعى عند النقطة التى سوف يتم عندها الدخول . لن يكون لجملة تأثير على منطق البرنامج الفرعى .

يمكن أن يكون هناك عدة جمل تعلن عن المداخل ENTRY فى البرنامج الفرعى . كل جملة تعرف نقطة دخول مختلفة . ومن الواضح أن كل نقطة دخول يجب أن يكون لها اسم خاص بها . وعلاوة على ذلك ، لا يستلزم أن تتفق المعاملات الملازمة لنقطة دخول مع معاملات البرنامج الفرعى ، ولا تستلزم أيضاً أن تتفق مع معاملات أى نقطة دخول أخرى . ولكن عندما تستخدم نقطة دخول معينة NAME فى البرنامج الداعى ، يجب أن تتفق قائمة خلاصات جملة النداء مع معاملات جملة ENTRY.

شكل جملة النداء لنقطة دخول فى برنامج فرعى مشابه لجملة النداء الخاصة ببرنامج فرعى SUBROUTINE أو FUNCTION فى كلا الحالتين عندما تستخدم اسم المدخل NAME بواسطة برنامج داع ، سيتم التنفيذ عند أول جملة منفذة تلى إعلان المدخل ENTRY فى البرنامج الفرعى .

وهيكل البرنامج التالى يوضح المفاهيم السابقة :

برنامج رئيسى

```
..................
CALL ABC(P, Q, R)
..................
CALL TWOA(E, F)
..................
CALL ONEA(S, M)
..................
STOP
END
```

برنامج فرعى

```
   SUBROUTINE ABC(X, Y, Z)
   ....................
   GO TO 10
   ENTRY ONEA(W, K)
   ....................
   GO TO 10
   ENTRY TWOA(
   ....................
10 RETURN
   END
```

(ب) رجوع متعدد (للبرامج الفرعية SUBROUTINE فقط)

عند مقابلة جملة الرجوع RETURN فى أى SUBROUTINE ينتقل التحكم عادة إلى أول جملة منفذة تالية لجملة CALL فى البرنامج الداعى . يسمح لنا الفورتران أيضاً بالرجوع إلى نقط مختلفة من البرنامج الداعى . ونوضح هذه الإمكانية باستخدام هيكل البرنامج التالى :

برنامج رئيسى

```
DIMENSION TEST(100)
......................
CALL GRADE(A, 30, B, C, 20, TEST)
......................
END
```

برنامج فرعى

```
SUBROUTINE GRADE(R, *, S, T, *, W)
DIMENSION W(100)
......................
RETURN 2
......................
RETURN
......................
RETURN 1
END
```

لاحظ أن هناك نجمتين فى قائمة المعاملات فى جملة تعريف البرنامج الفرعى SUBROUTINE لاحظ أيضاً أرقام الجمل 30 و 20 فى قائمة الخلاصات المناظرة لهاتين النجمتين على الترتيب . وتبعاً لذلك فجمل RETURN 1 و RETURN 2 تتناظر أرقام الجمل 30 و 20 على الترتيب . وعلى ذلك إذا قابلت RETURN 1 فى البرنامج الفرعى ، فسوف ينتقل التحكم إلى الجملة رقم 30 فى البرنامج الداعى ، وإذا قابلنا RETURN 2 فسوف ينتقل التحكم إلى الجملة رقم 20 فى البرنامج الداعى . من ناحية أخرى ، عندما يقابل RETURN فسوف يعود التحكم إلى أول جملة منفذة تالية لجملة CALL.

وبصورة عامة فإن جملة RETURN المتعددة لها الشكل :

RETURN *n*

حيث *n* ثابت صحيح بدون إشارة . تظهر هذه الجملة فقط فى برنامج فرعى به على الأقل عدد *n* من النجوم فى قائمة المعاملات الخاصة به وحيث تقابل كل نجمة رقم جملة فى قائمة الخلاصات . بعد ذلك عندما نقابل RETURN *n* فى البرنامج الفرعى فسوف ينتقل التحكم إلى الجملة التى رقمها مساو للرقم *n* فى قائمة الخلاصات فى البرنامج الداعى .

١١ — ٨ جمل COMMON غير المميزة

تذكر أن المتغيرات فى أى برنامج فرعى خاصة بهذا البرنامج الفرعى ما عدا المعاملات . وبالتالى فإن الوسيلة الوحيدة لتبادل المعلومات بين البرنامج الداعى والبرنامج الفرعى هى تحديدها فى قائمة علامات . ومع ذلك فيمكن أن يكون هذا التنظيم معقداً إذا كان عدد أماكن الذاكرة التى سوف تتقاسمها عدة برامج فرعية كبيرة جداً . يمدنا الفورتران بوسيلة بديلة يمكن البرامج الفرعية من اقتسام مساحات ذاكرة مشتركة . بالتحديد ، يمكن تعريف مساحة تخزين مشتركة باستخدام جملة COMMON سيكون لكل البرامج التى بها تعريف مناسب لهذه الجملة حق التوصل لأى بيانات مخزنة فى هذه المساحة .

هناك نوعان من جمل COMMON : COMMON مميزة و COMMON غير مميزة (خالية) . كلا النوعين جمل غير منفذة . ويجب أن توضع فى بداية أى برنامج قبل أى جملة منفذة . سنناقش جمل COMMON غير المميزة فى هذا القسم . وفى القسم التالى سنناقش جمل COMMON المميزة . وفيما يلى جمل COMMON غير المميزة (أى الخالية) النموذجية :

```
DIMENSION Y(2, 3), W(2)
COMMON X, Y, Z, W
```

لاحظ أن جملة COMMON تبدأ بكلمة COMMON وتتبعها قائمة من أسماء المتغيرات ربما يكون فيها أسماء مجموعات متراصه تفصل عن بعضها بواسطة فصلات . يمكن أن نتجنب استخدام جملة DIMENSION منفصلة للمجموعات المتراصة الموجودة فى جملة COMMON بتضمين حجم المجموعة المتراصة فى جملة COMMON نفسها . فمثلا كل من التنظيمات التالية :

```
COMMON X, Y(2, 3), Z, W(2)      or      DIMENSION Y(2, 3)
                                        COMMON X, Y, Z, W(2)
```

مساو لجملة الإعلان السابقة .

تحجز أى جملة من جمل COMMON المذكورة عالية كتلة مشتركة مكونة من أماكن ذاكرة عددها (2 + 1 + 6 + 1) حيث يتم تحديد أماكن الذاكرة تبعاً لترتيب قائمة المتغيرات . بالتحديد ، وفى هذا المثال ، تسمى أماكن الذاكرة كما فى شكل ١١ — ١ .

ونوضح هنا كيف تستخدم جملة COMMON وكيف تقتسم مساحة التخزين المشتركة بواسطة هيكل البرنامج المعطى فى شكل ١١ — ٢ . (لمجرد التبسيط فإن البرامج الفرعية تستخدم نفس جملة COMMON وبمنتهى الدقة ، رغم أن ذلك غير إلزامى . انظر ملاحظة ا على صفحة ٣١٩) . لاحظ أن البرنامج الرئيسى يعرف مساحة تخزين مشتركة تحتوى على 20 مكاناً من أماكن الذاكرة ويستدعى ثلاثة برامج فرعية وبرنامج دالة فرعياً .

X	
Y(1. 1)	
Y(2. 1)	
Y(1, 2)	
Y(2, 2)	
Y(1, 3)	
Y(2, 3)	
Z	
W(1)	
W(2)	

شكل ١١ — ١

البرنامج الرئيسى

```
COMMON A(10), B, X(2, 4), Q
.........................
CALL XXX(S, I)
.........................
CALL WWW(P1, P2, P3)
.........................
T = TAX + ZZZ(U, V)
.........................
CALL YYY
.........................
END
```

البرنامج الفرعى رقم ١

```
SUBROUTINE XXX(F, K)
COMMON A(10), B, X(2, 4), Q
.........................
RETURN
END
```

البرنامج الفرعى رقم ٢

```
SUBROUTINE YYY
COMMON A(10), B, X(2, 4), Q
.........................
RETURN
END
```

البرنامج الفرعى رقم ٣

```
FUNCTION ZZZ(G, H)
COMMON A(10), B, X(2, 4), Q
.........................
ZZZ = ...........
RETURN
END
```

البرنامج الفرعى رقم ٤

```
SUBROUTINE WWW(F, G, H)
B = 0.0
.........................
RETURN
END
```

شكل ١١ — ٢

تحتوى البرامج الفرعية XXX و YYY ودالة البرنامج الفرعى ZZZ على جملة COMMON ومن ثم ، سيكون له حق التوصل لمساحة التخزين المشتركة . وعموماً فكل برنامج من البرامج يستطيع أن يستخدم أو يغير أى قيمة مخزنة فى هذه المساحة . من ناحية أخرى ، حيث أن البرنامج الفرعى WWW لا يوجد به جملة COMMON فبالتالى ، ليس له حق التوصل لمساحة التخزين المشتركة .

توجد جملة واحدة أخرى نريد أن نؤكدها . وهى أن الجملة :

```
B = 0.0
```

فى البرنامج الفرعى WWW ليس لها تأثير على مساحة التخزين المشتركة حيث أن B خاصة فقط بالبرنامج الفرعى WWW ومع ذلك إذا ظهرت هذه الجملة فى أى من البرامج الفرعية الأخرى . فسوف تتغير قيمة B فى مساحة التخزين المشتركة وتصبح 0.0 بعد تنفيذها . علاوة على ذلك فستكون هذه القيمة B متاحة لأى برامج أخرى بعد ذلك .

ملاحظة ١ : استخدام جمل COMMON غير المميزة (الخالية) يخلق مساحة تخزين مشتركة واحدة فقط . يمكن التوصل إليها ببرامج فرعية مختلفة . ومع ذلك فأماكن الذاكرة فى هذه المساحة المشتركة يمكن أن تسمى بأسماء مختلفة فى البرامج الفرعية المختلفة . فمثلا ، نفرض أن البرنامج الفرعى XXX السابق له الشكل التالى :

```
SUBROUTINE XXX(F, K)
COMMON AA(4, 2), BB, CC(11)
.............................
RETURN
END
```

رغم أن جملة COMMON تشير إلى نفس مساحة التخزين المشتركة ، إلا أن أماكن الذاكرة فى هذه المساحة المشتركة للتخزين تسمى بأسماء مختلفة فى البرامج المختلفة كما هو موضح فى الشكل ١١ - ٣ . وبذلك إذا ظهرت الجملة :

CC(2) = 5.5

فى البرنامج الفرعى XXX وتم تنفيذها فى البرنامج الفرعى ، لتغيرت قيمة B أيضاً فى البرامج الفرعية YYY و ZZZ وفى البرنامج الرئيسى وأصبحت 5.5 .

A(1)		AA(1, 1)
A(2)		AA(2, 1)
	⋮	
A(8)		AA(4, 2)
A(9)		BB
A(10)		CC(1)
B		CC(2)
X(1, 1)		CC(3)
X(2, 1)		CC(4)
	⋮	
X(2, 4)		C(10)
Q		C(11)
↑ الاسم فى YYY ، ZZZ البرنامج الرئيسى		↑ الاسم فى XXX

شكل ١١ - ٣

ملاحظة ب : يمكن أن يكون عدد خلايا الذاكرة المعرفة فى المساحة المشتركة COMMON فى البرنامج الفرعى أصغر من المساحة المشتركة المعرفة فى البرنامج الرئيسى . فمثلا . مسموح بالتالى :

COMMON X, A(5), Y برنامج رئيسى
COMMON B(2, 2) برنامج فرعى

وعلاوة على ذلك ستقتسم المتغيرات خلايا الذاكرة فى المساحة المشتركة COMMON كما يلى :

الاسم فى البرنامج الرئيسى الاسم فى البرنامج الفرعى	X B(1, 1)	A(1) B(2, 1)	A(2) B(1, 2)	A(3) B(2, 2)	A(4)	A(5)	Y

لاحظ أن العناصر التى فى B(1, 1), X فى جمل COMMON تقتسم نفس المكان . ومن ناحية أخرى ، لا يمكن أن تتجاوز عدد خلايا الذاكرة فى المساحة المشتركة COMMON الموجودة فى البرنامج الفرعى تلك الموجودة فى البرنامج الرئيسى . فمثلا غير مسموح بما يلى :

COMMON C(2, 3) برنامج رئيسى
COMMON D(8) برنامج فرعى

ملاحظة ج : يمكن استخدام عدة جمل COMMON غير مميزة فى برنامج واحد . ومع ذلك فإنه يكافئ كتابة جملة COMMON واحدة طويلة . فمثلا :

COMMON A(10), T
COMMON C(2) تكافئ COMMON A(10), T, C(2)

ويتم حجز 13 مكاناً من أماكن الذاكرة للمساحة المشتركة COMMON.

١١ — ٩ جمل COMMON المميزة

تخلق جمل COMMON غير المميزة مساحة تخزين مشتركة واحدة فقط . وباستخدام جمل COMMON المميزة يمكن أن نعرف عدة مساحات تخزين مشتركة. يستدل على ذلك من اسم جملة COMMON فلكل من هذه المساحات المشتركة للتخزين عنوان (اسم) . وهذه الجمل غير منفذة أيضاً مثل جمل COMMON غير المميزة » .

فيما يلى نموذج لجملة COMMON المميزة :

COMMON /A/ TAX(10). X, INF(2, 4)

لا تعلن هذه الجملة أن TAX و INF أسماء مجموعات متراصة فقط ولكنها تعرف أيضاً مساحة تخزين مشتركة مسماة (لها عنوان) باسم A وبها 19 مكاناً من أماكن التخزين كما هو موضح فى شكل ١١ — ٤ . لاحظ أن جملة COMMON المميزة تبدأ بكلمة COMMON ويتبعها عنوان (اسم) الكتلة المشتركة محاطاً بشرطات مائلة . (ويتكون العنوان ، مثل أى اسم متغير آخر من عدد يصل إلى ستة حروف أبجدية مع ضرورة كون الحرف الأول أبجدياً) يلى بعد ذلك قائمة المتغيرات وأسماء المجموعات المتراصة المعطاة لأماكن الذاكرة فى هذه المساحة المشتركة للتخزين .

يمكن أن نعرف عدة مساحات تخزين مشتركة مميزة فى جملة COMMON واحدة . فمثلا :

COMMON /A/ TAX(10), X, INF(2, 4)/B/POP(100)

تعرف مساحتان مشتركتان . إحداهما باسم A والأخرى باسم B . ويتذكر القارئ أن المجموعات المتراصة يمكن أن تعرف إما فى جملة DIMENSION أو جملة COMMON ولكن ليس فى كليهما . وبذلك ، أى من الترتيبات الآتية يكافئ الجملة السابقة :

A

TAX(1)	
⋮	
TAX(10)	
X	
INF(1, 1)	
INF(2, 1)	
⋮	
INF(2, 4)	

شكل ١١ — ٤

```
DIMENSION TAX(10), INF(2, 4), POP(100)
COMMON /A/TAX, X, INF /B/POP
```

أو

```
DIMENSION TAX(10)
COMMON /A/TAX, X, INF(2, 4)
COMMON/B/POP(100)
```

وأننا لنؤكد عدم وجود فصلة تفصل بين المساحات المشتركة المميزة فى جملة COMMON الواحدة (فمثلا لا توجد فصلة تسبق /B/ فى أى من الجمل السابقة) .

الميزة الأساسية للمساحات المشتركة المسماة (المميزة) هى إعطاء الحق لبرامج فرعية مختلفة فى التوصل لمساحات مشتركة مختلفة . وأنه لغاية فى الأهمية أن نتذكر أن اسم الكتلة المشتركة هو الذى يتداول من برنامج فرعى إلى آخر . أما أسماء خلايا الذاكرة بداخل كتلة مشتركة معينة فهى محلية لهذا البرنامج الفرعى ، ومن ثم ، يمكن أن تكون مختلفة . كما تم مناقشتها سابقاً فى حالة الكتلة المشتركة غير المميزة . ونوضح ذلك بالمثال التالى .

مثال ١١ — ٢

ادرس هيكل البرنامج فى شكل ١١ — ٥ . لاحظ أنه تم تعريف مساحتين مشتركتين باسم AX و AY البرنامج الفرعى SUB1 له حق التوصل للمساحة المشتركة AY التى تحتوى على ثمانية أماكن للذاكرة وأسماء هذه الأماكن فى البرنامج الفرعى هى نفس الأسماء فى البرنامج الأساسى ، أى X(1, 1) و X(2, 1) و . . . و X(2, 4) ويقتسم البرنامج الفرعى SUB2 مع البرنامج الرئيسى المساحة المشتركة المسماة AX والتى بها 11 مكاناً للذاكرة . تسمى هذه الأماكن A(1) و . . . و A(10) فى البرنامج الرئيسى ولكن تسمى T(1, 1) و . . . و T(2, 3) و S(1) و . . . و S(5) فى البرنامج الفرعى كما هو موضح فى الشكل ١١ — ٦ .

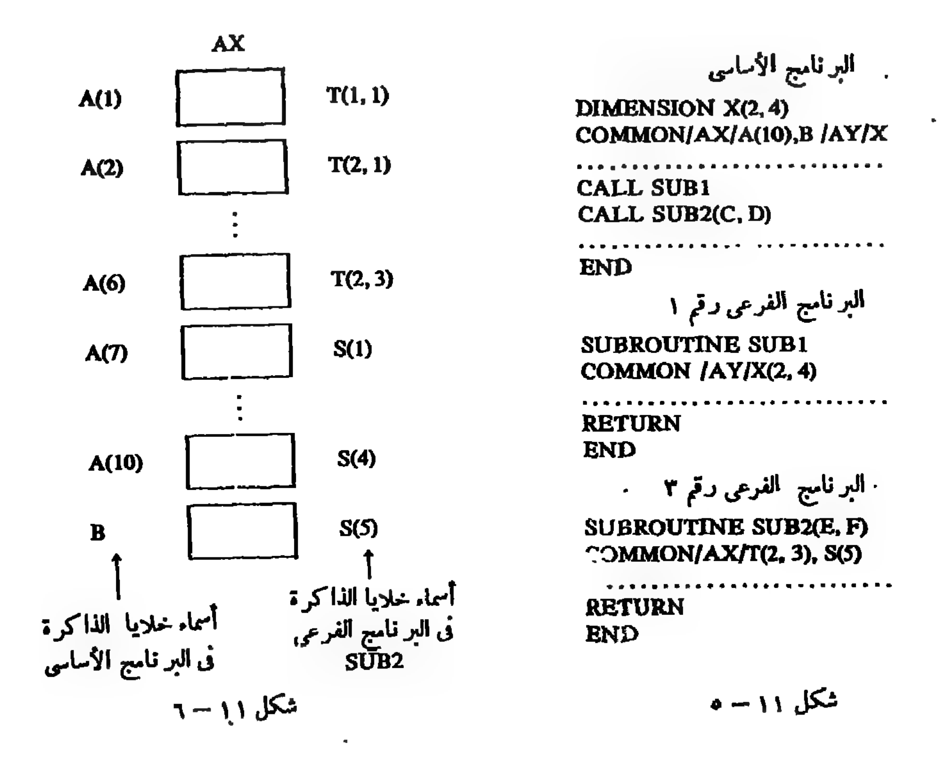
ونختم بملحوظات قليلة أخرى حول جمل COMMON المميزة :

١ – إذا تشابه اسم مساحتين مشتركتين فى برنامج فرعى ، كما يلى :

```
COMMON /A/T(2, 4)/B/X, Y, Z /A/Q(10)
COMMON /B/W(10)
```

فتكون النتيجة تجميعها ، أى يمكننا كتابة الجمل السابقة كما يلى :

```
COMMON /A/T(2, 4), Q(10) /B/X, Y, Z, W(10)
```



شكل ١١ – ٦

شكل ١١ – ٥

٢ – يجب أن يكون العدد الكلى لأسماء الذاكرة فى مساحة مشتركة مميزة هو نفس العدد فى كل البرامج الفرعية التى لها حق التوصل لهذه المساحة . لاحظ أن هذا يختلف عما سبق ذكره عن المساحة غير المميزة .

٣ – يمكن أن تعرف المساحات المشتركة المميزة وغير المميزة فى نفس البرنامج باستخدام نفس جملة COMMON فمثلا تخلق الجملة التالية :

```
COMMON TAX(100), A, B /Q/X(25, 10)
```

مساحة مشتركة غير مميزة بها عدد 102 خلية ذاكرة ومساحة مشتركة مميزة باسم Q بها عدد 250 خلية ذاكرة . ويمكن أن نستخدم الجملة التالية لتفى بنفس الغرض :

```
COMMON /Q/X(25, 10)//TAX(100), A, B
```

لاحظ عدم وجود اسم بين الشرطات المائلة بالنسبة لمساحة التخزين الثانية . ومن ثم تخلق مساحة مشتركة غير مميزة كما سبق ذكره . هذا هو السبب فى أن جملة COMMON غير المميزة تسمى أيضاً جملة COMMON الخالية .

يمكن أن يكون استخدام جمل COMMON المميزة وغير المميزة بصورة عامة معقداً للغاية ويؤدى للارتباك، لذلك فإننا نعتبر أن أى مناقشة لهذا الموضوع أكثر من ذلك تكون خارج نطاق هذا الكتاب .

١١ — ١٠ جمل EQUIVALENCE

افرض أن المبرمج أدرك بعد كتابة برنامجاً طويلا وبعد تثقيبه ، أنه استخدم أسماء مختلفة ، وليكن MAX و LARGE و JBIG لنفس الكمية ، فيمكن أن يضيف المبرمج ببساطة جملة EQUIVALENCE فى بداية البرنامج بدلا من تغيير كل الأسماء وإعادة تثقيب البطاقات .

EQUIVALENCE (MAX, LARGE, JBIG)

تخبر هذه البطاقة الحاسب أن الأسماء MAX و LARGE و JBIG تمثل كلها نفس أماكن الذاكرة .

لاحظ أن جملة EQUIVALENCE تبدأ بكلمة EQUIVALENCE وتتبع بأقواس تحيط بتلك المتغيرات التى تشير إلى نفس أماكن الذاكرة .

يمكن أن تستخدم جملة EQUIVALENCE أيضاً لتوفير أماكن الذاكرة وذلك بتخصيص أسماء مختلفة إلى نفس أماكن الذاكرة ، بشرط أن تستخدم المتغيرات المختلفة فى أجزاء مختلفة من البرنامج . فمثلا ، إذا أردنا كتابة برنامج لحساب المتوسط AVE والوسيط XMED والقيمة الكبرى BIG لعدد 100 رقم تم طباعتها . افرض أننا لن نحتاج المتغيرات AVE و XMED و BIG فى نفس الوقت فيمكن أن نوفر مساحة التخزين بتخصيص المتغيرات الثلاثة إلى نفس أماكن الذاكرة باستخدام جملة EQUIVALENCE

EQUIVALENCE(AVE, XMED, BIG)

وكما هو متبع ، فيمكن أن نستخدم جملة EQUIVALENCE واحدة لتحديد عدة مجموعات متساوية . فمثلا :

EQUIVALENCE(A, X, Z), (NEXT, LAST), (C, D, E, F)

تخبر الحاسب أن A و X و Z سوف تشترك فى نفس مكان التخزين ، وسوف تشترك NEXT و LAST فى نفس مكان التخزين وأن C و D و E و F سوف تشترك فى نفس مكان التخزين .

يحدث التطبيق الرئيسى لجملة EQUIVALENCE مع المجموعات المتراصة حيث يمكن توفير آلاف من أماكن التخزين . ومع ذلك يمكن أن تكون الطريقة التى تحتل بها المجموعات المتراصة نفس المساحة طريقة خادعة ، وبذلك فجمل EQUIVALENCE التى تشمل مجموعات متراصة ليست ببساطة الجمل الخاصة بالمتغيرات بدون دليل والتى قمنا بمناقشتها مسبقاً .

افرض مثلا أن A و B و C مجموعات متراصة فى برنامج وقد تم تعريفها بواسطة :

DIMENSION A(50, 100), B(25, 50), C(3750)

افرض أننا عرفنا بعد ذلك أن المجموعة المتراصة A مطلوبة فى بداية البرنامج فقط ولا تستخدم فى حسابات المجموعة المتراصة B و C نستطيع أن نوفر أماكن الذاكرة بتخزين المجموعات المتراصة B و C فى أماكن الذاكرة المحجوزة لـ A وبالتحديد نشير إلى أن أول عنصر من A وأول عنصر من B سيكون لهما نفس أماكن الذاكرة بواسطة :

EQUIVALENCE(A(1, 1), B(1, 1))

حيث أن عناصر المجموعات المتراصة متصلة فإن هذه الجملة تخبر الحاسب أيضاً أن أول 1250 مكان ذاكرة من A ستحتل نفس الأماكن مثل 1250 مكان من B (هل تستطيع أن تحدد التناظر ؟) يمكن أن تستخدم 3750 مكاناً الباقية من A للمجموعة المتراصة C بواسطة الجملة التالية :

```
EQUIVALENCE(A(1, 26), C(1))
```

يمكن أن تدمج هاتين الجملتين فى جملة واحدة كما يلى :

```
EQUIVALENCE(A(1, 1), B(1, 1)), (A(1, 26), C(1)),
```

وتتداخل الآن مساحات التخزين ويتم حجز 5000 مكان من الذاكرة فقط بدلا من 10,000 مكان . الشئ القاطع الذى يجب أن نتذكره حول المجموعات المتراصة هو أن :

١ – عناصر المجموعة المتراصة وتعطى دائماً جنباً إلى جنب فى ترتيب خطى، وهذا لا يمكن تغييره بأى جملة EQUIVALENCE أخرى .

٢ – يمكن أن تظهر عناصر المجموعة المتراصة فقط فى جملة EQUIVALENCE وليس اسم المجموعة المتراصة نفسها .

ملاحظة : تذكر أن المجموعات المتراصة تخزن خطياً فى الذاكرة رغم أن المجموعات المتراصة يمكن أن تكون متعددة الأبعاد . تسمح بعض المترجمات باستخدام المكان الخطى لعناصر أى مجموعة متراصة فى جملة EQUIVALENCE

مثلاً ، تحتل A(1, 1) المكان الأول فى المجموعة المتراصة A السابقة وتحتل A(1, 26) المكان 1251 من A . وتشغل أيضاً B(1, 1) المكان الأول فى B وبالتالى فيمكن استبدال الجملة :

```
EQUIVALENCE(A(1, 1), B(1, 1)), (A(1, 26), C(1))
```

بواسطة الجملة :

```
EQUIVALENCE(A(1), B(1)), (A(1251), C(1))
```

والميزة فى استخدام المكان الخطى هو سهولة حساب المكان .

فى أى من الأشكال السابقة يجب أن نذكر أن الأدلة لابد وأن تكون قيما ثابتة إذ أن جملة EQUIVALENCE غير منفذة ، وتحجز أماكن التخزين أثناء الترجمة .

نعطى بعد ذلك أمثلة أخرى تشمل مجموعات متراصة مع جمل EQUNALENCE .

مثال ١١ – ٣

(ا) ادرس الجمل التالية :

```
DIMENSION A(2, 3)
EQUIVALENCE(A(2), X, Y), (A(5), C)
```

تشارك المتغيرات أماكن التخزين كما يلى :

A(1, 1)	A(2, 1) X Y	A(1, 2)	A(2, 2)	A(1, 3) C	A(2, 3)

أى ، يخصص لكل من A(2, 1) و X و Y نفس أماكن التخزين ، ويخصص لكل من A(1, 3) و C نفس أماكن التخزين .

(ب) افرض إمكانية استخدام المكان الخطى لمجموعة متراصة . ادرس الجمل التالية :

```
DIMENSION A(2, 3), B(2, 2), C(5)
EQUIVALENCE(A(1), B(1), C(1))
```

تشارك المجموعات المتراصة مساحة التخزين كما يلى :

A(1, 1) B(1, 1) C(1)	A(2, 1) B(2, 1) C(2)	A(1, 2) B(1, 2) C(3)	A(2, 2) B(2, 2) C(4).	A(1, 3) C(5)	A(2, 3)

لاحظ أن C(4) أجبرت على مشاركة المكان مع A(2, 2) و B(2, 2) حيث تم تخصيص أول عنصر فى كل مجموعة متراصة إلى نفس المكان .

(جـ) ادرس الجمل التالية :

```
DIMENSION A(2, 3), C(7)
EQUIVALENCE(A(3), C(1))
```

فسوف تظهر المجموعات المتراصة فى مساحة التخزين كما يلى :

A(1, 1)	A(2, 1)	A(1, 2) C(1)	A(2, 2) C(2)	A(1, 3) C(3)	A(2, 3) C(4)	C(5)	C(6)	C(7)

لاحظ أن المجموعة المتراصة C ترحل من الجانب الأيمن .

نستطيع أن نستخدم جملة EQUIVALENCE مع جملة COMMON ومع ذلك ، فيمكن لجملة EQUIVALENCE أن تزيد من طول الخزن المشترك فقط ، ولكن لا يمكنها أن تغير الأماكن الأصلية للمتغيرات . نوضح ذلك بأمثلة .

مثال ١١ - ٤

(ا) ادرس الجمل التالية :

```
DIMENSION A(4), C(2), D(8)
COMMON A, B, C
EQUIVALENCE(A(3), D(1))
```

ستظهر مساحة التخزين المشتركة كما يلى :

```
A(1)  A(2)  A(3)  A(4)  B     C(1)  C(2)
            D(1)  D(2)  D(3)  D(4)  D(5)  D(6)  D(7)  D(8)
```

(لم نرسم صناديق مع أسماء المتغيرات كما فعلنا فى مثال ١١ — ٣ وذلك لسهولة الترميز) تخلق جملة COMMON الأصلية سبعة أماكن فى المخزن المشترك . تزيد جملة EQUIVALENCE مساحة المخزن المشترك إلى 10 أماكن .

(ب) جمل FORTRAN التالية غير مسموح بها :

```
DIMENSION A(4), C(2), D(8)
COMMON A, B, C
EQUIVALENCE(A(1), D(3))
```

وللأسباب السابقه يجب أن يبدو اقتسام المخزن المشترك كالتالى :

```
            A(1)  A(2)  A(3)  A(4)  B     C(1)  C(2)
D(1)  D(2)  D(3)  D(4)  D(5)  D(6)  D(7)  D(8)
```

تمثل (A(1 هنا المركز الثالث ، ولكن تبعاً لجملة COMMON يجب أن تحتل (A(1 المركز الأول حيث أن الفورتران لا يسمح لنا بتغيير الأماكن الأصلية للمتغيرات فى المخزن المشترك ، لذلك فإن هذه الجمل غير مسموح بها .

يمكن أن تكون الاستخدامات الأخرى لجملة EQUIVALENCE وبخاصة التى تشمل جمل COMMON معقدة جداً وتعتمد على الآلة ؛ من ثم ، فتعتبر أى مناقشة أخرى خارج نطاق هذا الكتاب . وعلاوة على ذلك ، نظراً لأنه من الممكن الاستغناء عن جملة EQUIVALENCE كلية فإننا ننصح المبرمج المبتدئ أن يتجنب استخدام جملة EQUIVALENCE كلما أمكن ذلك .

١١ — ١١ كتلة البيانات

لا تسمح بعض المترجمات باستخدام جملة DATA لإعطاء قيم ابتدائية لمتغيرات فى مساحات COMMON المميزة ، ولا يسمح البعض بإعطاء قيم ابتدائية لأى مساحات COMMON إلا أنه يمكن أن تستخدم برنامجاً فرعياً يسمى BLOCK DATA لهذا الغرض .

يمكن أن يتكون البرنامج الفرعى BLOCK DATA من الجمل غير المنفذة التالية فقط :

```
DATA
COMMON
DIMENSION
EQUIVALENCE
```

وأى إعلانات عن النوع . وسوف نوضح كيفية عمل مثل هذا البرنامج الفرعى بمثال

افرض أن لدينا الآتى فى برنامج رئيسى :

```
DIMENSION X(20), Y(15)
COMMON /A/X, U, V /B/Y
          :
END
```

افرض أننا نريد إعطاء قيمة ابتدائية 0.0 لكل عنصر من X و Y وحيث أن X و Y فى مساحات مشتركة COMMON مميزة فإن استخدام جملة DATA غير مسموح بها . ولكننا نستطيع أن نستخدم جمل تخصيص لإعطاء القيم الابتدائية التى سنحتاجها وقت التنفيذ . من ناحية أخرى يمكن أن نستخدم البرنامج الفرعى BOLCK DATA الآتى :

```
BLOCK DATA
DIMENSION X(20), Y(15)
COMMON /A/X, U, V /B/Y
DATA X/20*0.0/, Y/15*0.0/
END
```

نلخص فيما يلى استخدام البرنامج الفرعى BLOCK DATA :

١ – يبدأ البرنامج الفرعى دائماً بالجملة BLOCK DATA .

٢ – يجب أن نذكر كل المتغيرات الموجودة فى المساحة COMMON فى البرنامج الفرعى برغم أننا نريد أن نعطى قيمة ابتدائية لبعض المتغيرات فقط .

٣ – يمكن أن يستخدم برنامجاً فرعياً BLOCK DATA واحداً فقط لإعطاء قيم ابتدائية لمتغيرات فى أى مساحة مشتركة COMMON واحدة .

١١ – ١٢ خارجى EXTERNAL

نذكر أن الاتصال الوحيد بين البرنامج الداعى وأى برنامج فرعى (إلى جانب COMMON و EQUIVALENCE) هى عن طريق قائمة الخلاصات . يكون مستحباً فى بعض الأحيان أن نستخدم اسم برنامج فرعى كخلاصة حقيقية فى جملة الاستدعاء لبرنامج فرعى آخر . هيكل البرنامج التالى يوضح مثل هذا الموقف :

البرنامج الأساسى

```
           ............
           EXTERNAL TOTAL
           X = BAL(ASSET, PAY, TOTAL)
           ............
برنامج فرعى { END
           FUNCTION BAL(A, B, C)
           T = C(A)
           ............
           RETURN
برنامج فرعى { END
           FUNCTION TOTAL(Z)
           ............
           RETURN
           END
```

كما هو مشار إليه بأحد الأسهم ، فاسم البرنامج الفرعى TOTAL يتم نقله إلى البرنامج الفرعى BAL باستخدام المعامل C . لذلك ، تحسب C(A) فى BAL باستخدام البرنامج الفرعى TOTAL كما هو مشار إليه بالسهم الآخر .

ولذلك فبدون إعلان خاص ، لن يتمكن المترجم من التفرقة بين اسم البرنامج الفرعى واسم متغير بسيط يحمل قيمة . ولتجنب التشويش يستخدم إعلان EXTERNAL فى البرنامج الداعى لتشير أن اسم البرنامج الفرعى المستخدم كخلاصة هو خارجى بالنسبة لهذا البرنامج .

مسائل

جمل IMPLICIT وجمل النوع TYPE

١١ — ١ اكتشف الأخطاء ، إن وجدت ، فى كل جزء برنامج مما يأتى :

```
INTEGER, RATE, TIME, DISTANCE
(ا) DIMENSION R(4), S(5, 3)
(ب) DOUBLE PRECISION A(7), B, R(8), S,
(جـ) IMPLICIT, COMPLEX (A - D, F, G) LOGICAL(C, X - Z)
(د) IMPLICIT REAL M, N, COMPLEX P - R
(هـ) DOUBLE PRECISION NUMBER(4, 7) X, Y, TEST(8, 15) L, M
```

١١ — ٢ ادرس جزء البرنامج التالى :

```
IMPLICIT COMPLEX(H - K) LOGICAL(P, R)
REAL COEF, KAPPA, LAMBDA
INTEGER RATE, TIME
```

حدد نوع كل متغير مما يلى :

(ا) ALPHA (ب) INDEX (جـ) KAPPA

(د) KOUNT (هـ) PARTY (و) QUEUE

(ز) LAST (ح) LAMBDA (ط) RATE

(ى) ROOM (ك) TIME (ل) TARGET

دقة متضاعفة

١١ — ٣ حدد ما إذا كان كل متغير يحتل ثابتاً حقيقياً (دقة مفردة) أو دقة متضاعفة (مع فرض سبع خانات محتوية الثوابت الحقيقية) :

(ا) 1.45D + 03 (ب) 5.9321E − 7 (جـ) 3456.765432

(د) 12345678 (هـ) 23.4 + 0.D0 (و) 111.222333E − ٤

(ز) 123.123123 + 0.E0 (ح) 5.55E − 26 + 58.4 (ط) 6.7 + 4.5D1 − 7.8E3

١١ — ٤ حدد الخرج لكل جزء من البرامج التالية :

(ا)

```
   DOUBLE PRECISION DA
   A = 0.555666777888999
   DA = 0.555666777888999
   WRITE(6, 10) A, A
10 FORMAT(1X, E25.3, 5X, E25.9)
   WRITE(6, 20) DA, DA
20 FORMAT(1X, D25.3, 5X, E25.9)
   STOP
   END
```

(ب)

```
   DOUBLE PRECISION X, Y, Z
   X = 70/9
   Y = 70.0/9.0
   Z = 70.0D0/9.0D0
   WRITE(6, 30) X, Y, Z
30 FORMAT(1X, D20.12)
```

١١ — ٥ افرض أن A = 9999999.0, B = 125.0, and C = 9999998.0 ولاحظ أن

X = A + B − C = 126.0

(حيث يمكن الحصول عليها باستخدام حساب الدقة المتضاعفة) .

(أ) احسب قيمة X باستخدام حساب الدقة المفردة (مع فرض سبع خانات معنوية)

(ب) أوجد الخطأ وكذلك نسبة الخطأ المئوية .

١١ - ٦ اكتب البرنامج الذى يحسب مضروب n أى $(n!)$ لـ $n = 1, 2, \ldots\ldots, 15$ (يجب أن يستخدم حساب الدقة المتضاعفة حيث أن كلا من الحساب الصحيح وحساب الدقة المفردة ليس بالدقة الكافية) .

١١ - ٧ تذكر أنه يمكن حساب مساحة المثلث ذى الأطوال a و b و c بواسطة :

$$\text{AREA} = \sqrt{s(s-a)(s-b)(s-c)}$$

حيث :

$$s = (a + b + c)/2$$

اكتب البرنامج الذى يقرأ قيم a و b و c ويحسب ويطبع المساحة AREA باستخدام حساب الدقة المتضاعفة .

أعداد مركبة

١١ - ٨ اجعل $z = 2 - 3i$ و $w = 4 + 5i$ أوجد ما يأتى جبرياً :

(أ) $z + w$ (ب) zw (جـ) z/w (د) $\bar{z}$ and $\bar{w}$ (هـ) $|z|$ and $|w|$

تحقق من النتائج على الحاسب باستخدام برنامج فورتران :

١١ - ٩ بفرض أن X و Y و Z متغيرات مركبة و A و B و C متغيرات حقيقية ، اكتشف الأخطاء إن وجدت فى كل جملة مما يأتى :

(أ) X = (3, 4.5)
(ب) Y = (A, B + C)
(جـ) Z = 3.6 + (5.0, 7.6)
(د) A = COMPLEX(B, C)

١١ - ١٠ افرض أنه تم تثقيب أول بطاقتين للبيانات كالتالى :

```
             1         2
    1234567890123456789012345
البطاقة الأولى     1.1  2.2  3.3  4.4
البطاقة الثانية    5.5  6.6  7.7  8.8
```

أوجد الخرج لكل جزء برنامج مما يلى :

(أ)

```
      COMPLEX A, B, C
      READ(5, 10) A, B
10    FORMAT(2F5.1)
      C = A*B
      WRITE(6, 20) C
20    FORMAT(1X, F10.2)
```

(ب)

```
      COMPLEX A, B, C
      READ(5, 50) A, B
50    FORMAT(3F5.1)
      C = A*B
      WRITE(6, 60) C
60    FORMAT(1X, 'REAL PART IS', 7X, F10.2/1X,
     1       'IMAGINARY PART IS', 2X, F10.2)
```

١١ – ١١ اكتب برنامج يقرأ ثلاثة أرقام مركبة ويطبع الأرقام الثلاثة على سطر ثم يطبع مجموعها وحاصل ضربها على سطر آخر .

١١ – ١٢ اكتب برنامجاً يقرأ الأرقام الحقيقية a و b و c ويحل معادلة الدرجة الثانية التالية :

$$ax^2 + bx + c = 0$$

باستخدام الصيغة الرياضية :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

(لاحظ أن الجذور تكون مركبة إذا كان مميز المعادلة $D = b^2 - 4ac$ أقل من صفر) .

جمل GO TO و ASSIGN والمداخل والعودة المتعددة

١١ – ١٣ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل الفورتران التالية :

(ا) ASSIGN +45 TO NUMBER
(ب) ASSIGN N TO NUMBER
(جـ) GO TO NUMBER (5, 78, 6, −43)
(د) GO TO 200, (75, 25, 200, 50)

١١ – ١٤ اكتشف الأخطاء ، إن وجدت ، فى كل جزء من برامج الفورتران التالية :

(ا)
```
ASSIGN 25 TO NEXT
GO TO NEXT, (50, 75, 300)
```

(ب)
```
ASSIGN 25 TO NEXT
NEXT = 50
GO TO NEXT, (25, 40, 50, 70)
```

١١ – ١٥ اكتشف الأخطاء إن وجدت ، فى هيكل ":" نامج التالى :

البرنامج الفرعى
```
)UTINE BIG(A, B, C)
.       ................
E        MIDD:.E(X, Y, Z)
..       ................
EN       :ATE(C, D, K)
.....    .............
END
```

البرنامج الرئيسى
```
..................
CALL BIG(X, Y, Z)
..................
CALL MIDDLE(P, A + B)
..................
CALL RATE(X + Y, J, N + 4)
..................
STOP
END
```

١١ – ١٦ ادرس جملة ALL" ذ"تية وكذا جملة SUBROUTINE المناظرة :

```
CALL UPDATE(A, 88, B, 36, C, N, 44)
SUBROUTINE UPDATE(X, *, Y, *, Z, NUMBER, *)
```

أوجد النقطة التى ينتقل إليها التحكم إذا قابلنا الجملة التالية فى البرنامج الفرعى UPDATE :

(ا) RETURN 1 (ب) RETURN 2 (جـ) RETURN 3 (د) RETURN

COMMON و EQUIVALENCE

١١ – ١٧ اكتشف الأخطاء ، إن وجدت ، فى كل جملة من جمل فورتران التالية :

(ا) COMMON (X, Y, Z)
(ب) EQUIVALENCE X, Y, Z
(جـ) COMMON X, Y(2, 4), Z(8), W,
(د) EQUIVALENCE (X, Y) (A(1), B(4), Z)
(هـ) COMMON /A/ NUM(5), X, /B/ INDEX(4, 5)

١١ – ١٨ ارسم شكل المخزن الذى ينتج من هيكل البرنامج التالى :

برنامج رئيسى

```
COMMON Y(4), A, B, C(2)
....................
CALL NNN(P, Q, R)
....................
END
```

برنامج فرعى

```
SUBROUTINE NNN(P1, P2, P3)
COMMON D, X(2), E(2), F(2)
...........................
END
```

١١ – ١٩ حدد شكل المخزن الذى ينتج من أجزاء البرنامج التالية :

(ا)

```
DIMENSION A(5), B(4), C(7)
EQUIVALENCE (A(1), B(1), C(2))
```

(ب)

```
DIMENSION X(2, 3), Y(3, 2)
EQUIVALENCE(X(3), Y(1))
```

١١ – ٢٠ افرض أن A مصفوفة أى مجموعة متراصة (3 × 3) ونريد أن نجعل DIAG مجموعة متراصة خطية بها ثلاثة عناصر تحدد مداخل القطر لـ A ناقش الخطأ فى جزء البرنامج التالى :

```
DIMENSION A(3, 3), DIAG(3)
EQUIVALENCE (A(1), DIAG(1)), (A(5), DIAG(2)), (A(9), DIAG(3))
```

اجابات المسائل المختارة

١٠ – ١ (ا) يجب ألا تكون هناك فصلة بعد INTEGER .

(ب) تم تعريف R مرتين بأحجام مختلفة . يجب ألا تكون هناك فصلة بعد S .

(جـ) يجب ألا تكون هناك فصلة بعد IMPLICIT يجب أن تكون هناك فصلة قبل LOGICAL .

(د) IMPLICIT REAL (M, N) و COMPLEX (P — R)

(هـ) يجب أن تكون هناك فصلة قبل X وفصل قبل L .

١١ – ٢

(ا) حقيقى (ب) مركب (جـ) حقيقى (د) مركب (هـ) منطقى (و) حقيقى

(ز) صحيح (ح) حقيقى (ط) صحيح (ى) منطقى (ك) صحيح (ل) حقيقى

١١ - ٣

(ا) مضاعف (ب) حقيق (جـ) مضاعف (د) لا هذا ولا ذاك ، حيث أنه صحيح

(هـ) مضاعف (و) مضاعف (ز) مضاعف (ح) حقيق (ط) مضاعف

١١ - ٤ (ا) 0.555666700E 00 0.556E 00 (ب) 0.700000000000D 01
0.555666778D 00 0.556D 00 0.777777700000D 01
0.777777777778D 01

١١ - ٥ (ا) 1.2.0 (ب) خطأ = 4 ، وهى 3 فى المائة .

١١ - ٧ يجب أن تعرف كل المتغيرات DOUBLE PRECISION ويجب أن تستخدم دالة الجذر التربيعى ذات الدقة المتضاعفة DSQRT :

```
      DOUBLE PRECISION A, B, C, S, AREA
      READ(5, 10) A, B, C
10    FORMAT(3D20.10)
      S = (A + B + C)/2.0
      AREA = DSQRT(S*(S - A)*(S - B)*(S - C))
      WRITE(6, 20) AREA
20    FORMAT(1X, 'AREA IS', 3X, D20.12)
      STOP
      END
```

١١ - ٨ (ا) $6 + 2i$ (ب) $23 - 2i$ (جـ) $-7/41 - 22i/41$ (د) $2 + 3i, 4 - 5i$ (هـ) $\sqrt{13}, \sqrt{41}$

١١ - ٩ (ا) يجب أن تكتب 3 فى الصورة 3.0

(ب) CMPLX (A, B + C) وليس (A, B + C)

(جـ) صواب.

(د) CMPLX (B, C) وليس COMPLEX (B, C)

١١ - ١٠ (ا) -8.47 (ب) -8.47 REAL PART IS
19.36 13.31 IMAGINARY PART IS

١١ - ١٣ (ا) يجب أن تكون 43 بدون إشارة لكونها رقم جملة .

(ب) يجب أن تكون N رقم جملة حقيقياً ، وليس متغيراً صحيحاً .

(جـ) يجب أن تكون هناك فصلة بعد NUMBER . 43— غير مسموح بها أيضاً .

(د) يجب أن تكون أول 200 متغير صحيح ، وليس ثابتاً صحيحاً .

١١ - ١٤ (أ) يجب أن تظهر القيمة 25 لـ NEXT فى القائمة التالية لـ NEXT فى جملة GO TO

(ب) يمكن تغيير قيمة NEXT بواسطة جملة ASSIGN أخرى فقط ، مثلا :

ASSIGN 30 TO NEXT

لا يمكن تغيره بواسطة جملة تخصيص حسابية .

١١ - ١٥ (أ) MIDDLE فى البرنامج الرئيسى لها خلاصتان فقط ولكن MIDDLE لها ثلاثة معاملات فى البرنامج الفرعى . يجب أن تكون الخلاصة الثانية J فى RATE حقيقية حيث أنها تناظر المعامل الحيق D .

١١ - ١٦ (أ) إلى الجملة رقم 88 فى البرنامج الداعى

(ب) إلى الجملة رقم 36 فى البرنامج الداعى

(جـ) إلى الجملة رقم 44 فى البرنامج الداعى

(د) إلى أول جملة منفذة تلى جملة الاستدعاء CALL .

١١ - ١٧ (أ) يجب ألا تكون المتغيرات محاطة بأقواس .

(ب) يجب أن تكون المتغيرات محاطة بأقواس .

(جـ) يجب ألا تكون هناك فصلة بعد W .

(د) يجب أن تكون هناك فصلة بعد (X, Y) .

(هـ) يجب ألا تكون هناك فصلة بعد X .

١١ - ١٨

الاسم فى البرنامج الرئيسى : Y(1) Y(2) Y(3) Y(4) A B C(1) C(2)

الاسم فى البرنامج الفرعى : D X(1) X(2) E(1) E(2) F(1) F(2)

١١ - ١٩

(أ)

A(1) A(2) A(3) A(4) A(5)
B(1) B(2) B(3) B(4)
C(1) C(2) C(3) C(4) C(5) C(6) C(7)

(ب)

X(1, 1) X(2, 1) X(1, 2) X(2, 2) X(1, 3) X(2, 3)
Y(1, 1) Y(2, 1) Y(3, 1) Y(1, 2) Y(2, 2) Y(3, 2)

١١ - ٢٠ تشير المتساويات الثلاثة إلى أن DIAG (1) يجب أن تقتسم مكان التخزين مع A(1, 1) و DIAG(2) مع A(2, 2) و DIAG(3) مع A(3, 3) كما يلى :

A(1, 1)	A(2, 1)	A(3, 1)	A(1, 2)	A(2, 2)	A(3, 2)	A(1, 3)	A(2, 3)	A(3, 3)
DIAG(1)				DIAG(2)				DIAG(3)

إلا أن ما سبق ينقض الشرط فى أن عناصر المجموعة المتراصة يجب أن تظهر جنباً إلى جنب فى المخزن ، أى ، أننا لا نستطيع أن نفصل عناصر DIAG

الفصل الثانى عشر

الفـورتران الهيـكلى

قد يحتاج التمرن فى عملية البرمجة لأكثر من عدة سنوات من التدريب والخبرة . وفى بعض الأحيان يصاب المبتدئون بالارتباك والأحباط عندما يتضح لهم أن برامجهم مجرد مجموعة من الجمل بدون اى تنظيم أو منطق واضح .

يمكن أن نعزى منبع الارتباك والخطأ الأساسى فى البرمجة بالفورتران إلى : (١) هياكل التحكم المربكة فى لغة الفورتران نفسها ، و(٢) الاستخدام غير المنظم لبعض هياكل الفورتران ، وبالأخص جمل GO TO .

وعبر السنوات برزت عدة قواعد لممارسة جيدة للبرمجة أدت إلى فكرة البرمجة الهيكلية . وتبعاً لذلك فقد ضمنت كثير من التركيبات الهيكلية الدقيقة بداخل عدة مشغلات للفورتران . وقد ساعدت تركيبات الفورتران الهيكلية هذه فى تخفيف ارتباك المبتدئين وتقليل نسبة الأخطاء ، فالبرامج المكتوبة بهذه الطريقة أكثر قابلية للقراءة ويمكن فهمها وتعديلها بسهولة أكثر .

يناقش هذا الفصل بعضاً من هياكل التحكم الشهيرة واستخدامها وعلاقتها بالفورتران غير الهيكلى . فى الحقيقة فالقراء الذين لهم حق استخدام مشغلات الفورتران الهيكلية يمكنهم دراسة IF الهيكلية بعد الفصل الرابع وحلقة DO الهيكلية بعد الفصل الخامس .

١٢ — ١ IF الهيكلية

هياكل IF فى الفورتران الهيكل لها تنوعات عديدة . ومع ذلك فهى تبدأ دائماً بكتلة جملة IF التى لها الشكل التالى :

IF(*logexp*) THEN

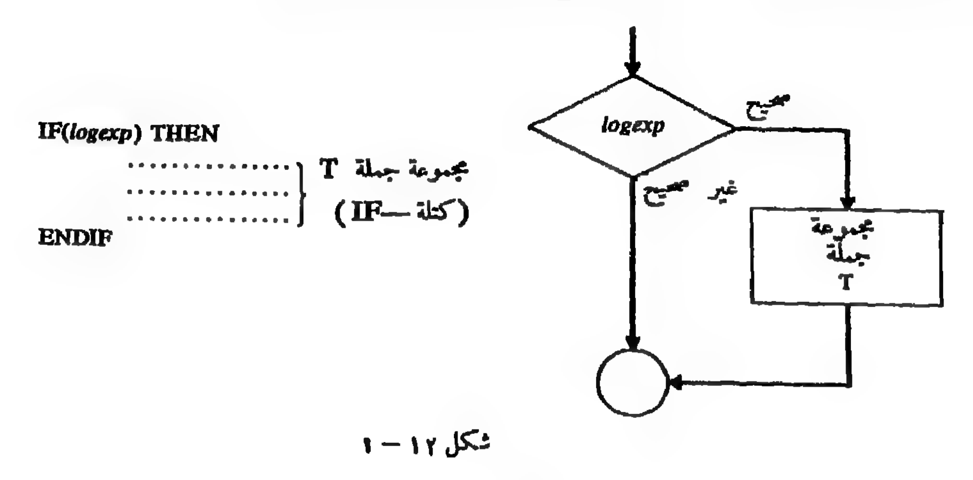
(حيث *logexp* هو تعبير منطق) وينتهى بجملة

ENDIF

تستخدم هاتان الجملتان كحدود لجسم الهيكل . وسوف نناقش بعضاً من هذه التنوعات .

البـديل المفـرد

يوضح شكل ١٢ — ١ أحد أشكال — IF الهيكلية مع خريطة سير العمليات الخاصة بها .



شكل ١٢ — ١

أولا تحسب قيمة التعبير المنطقى *logexp* كما هو موضح فى خريطة سير العمليات . إذا كان *logexp* صحيحاً فسوف تنفذ مجموعة جمل T بأكلها (وتسمى كتلة —IF) . أما إذا كان *logexp* غير صحيح ، فسوف تتخطى مجموعة جمل T بأكلها ، وينتقل التحكم إلى أول جملة تلى جملة ENDIF .

يكافىء هيكل —IF السابق أياً من أجزاء برامج الفورتران غير الهيكلية التالية والتى تستخدم جملة —IF المنطقية .

```
IF(logexp) GO TO 20
GO TO 10
20  ...........
    ...........   } مجموعة جمل T
    ...........
10  ...............
```

```
IF(.NOT.logexp) GO TO 10
    ...............
    ...............   } مجموعة جمل T
    ...............
10  ...............
```

لاحظ أن أول جزء يستخدم مكمل التعبير المنطقى (*logexp*) فى جملة IF المنطقية ، وهذا يجنب استخدام جملة GO TO إضافية كما فى الجزء الثانى . يمكن أن نحصل على مكمل التعبير المنطقى ببساطة بوضع .NOT. قبل التعبير . ومع ذلك فمكملات معاملات الترابط الفردية توضح كالآتى :

الأداة	.EQ.	.NE.	.GT.	.GE.	.LT.	.LE.
المكمل	.NE.	.EQ.	.LE.	.LT.	.GE.	.GT.

تذكر أن مكمل المكمل لتعبير منطقى *logexp* هو التعبير المنطقى *logexp* نفسه ، فمثلا (.NOT. *logexp*) .NOT. تكافىء *logexp* .

مثـال ١٢ – ١ :

أجزاء برامج الفورتران التالية متكافئة :

```
(أ)  IF(A.GT.B) THEN
        TEMP = A
        A = B
        B = TEMP
     ENDIF
```

```
(ب)  IF(A.LE.B) GO TO 10
        TEMP = A
        A = B
        B = TEMP
10   .....................
```

يكافىء هيكل —IF ذو البديل المفرد والذى يحتوى على جملة واحدة فقط ، جملة IF المنطقية ، والعكس صحيح ، فمثلا كل ما يأتى متكافىء :

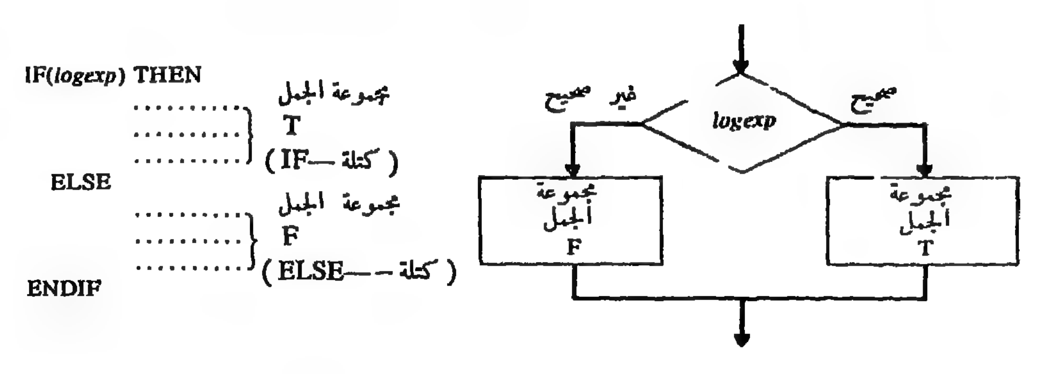
```
(أ)  IF(BIG.LT.A) THEN
        BIG = A
     ENDIF
```

```
(ب)  IF(BIG.LT.A) BIG = A
```

ملاحظة : لايسمح بنقل التحكم إلى داخل كتلة —IF من أى مكان خارج الكتلة نفسها .

البــديل الثنــائى

يمكن أن تضاف جملة ELSE فى هيكل —IF السابق وذلك لتنفيذ قرارات متعلقة ببديلين مختلفين ، يوضح شكل ١٢-٢ الشكل العام لمثل هذا الهيكل وخريطة سير العمليات الخاصة به .



شكل ١٢ – ٢

يتم أولا حساب التعبير المنطقى *logexp* كما هو موضح بخريطة سير العمليات ، إذا كان التعبير المنطقى *logexp* صحيحاً تنفذ كتلة IF (مجموعة جمل T) بأكملها . أما إذا كان التعبير المنطقى *logexp* غير صحيح ، فسوف تنفذ مجموعة جمل F (وتسمى كتلة —ELSE) بدلا منها . وفى كلتا الحالتين ، بعد الانتهاء من تنفيذ ، إما المجموعة T أو المجموعة F ، فسوف ينتقل التحكم إلى أول جملة منفذة بعد جملة النهاية ENDIF .

يكافىء هيكل —IF السابق أى من أجزاء برامج الفورتران التالية باستخدام ، جملة IF المنطقية :

(أ)

```
      IF(.NOT.logexp) GO TO 10
      ..........
      ..........   مجموعة جمل T
      ..........
      GO TO 99
10    ..........
      ..........   مجموعة جمل F
      ..........
99    ................
```

(ب)

```
      IF(logexp) GO TO 20
      ..........
      ..........   مجموعة جمل F
      ..........
      GO TO 99
20    ..........
      ..........   مجموعة جمل T
      ..........
99    ................
```

لاحظ أنه فى حالة عدم استخدام مكمل التعبير المنطقى *logexp* كما فى (ب) فإن أماكن مجموعات الجمل سوف تنعكس ، أى ، تظهر مجموعة F أولا ثم بعد ذلك مجموعة T بينما تظهر مجموعة T أولا فى هيكل IF ثم بعد ذلك مجموعة F .

مثال ١٢ – ٢

مايلى جزءان متكافئان كل يحسب الأجر GROSS لموظف يتقاضى أجرا ونصف إذا عمل ساعات إضافية :

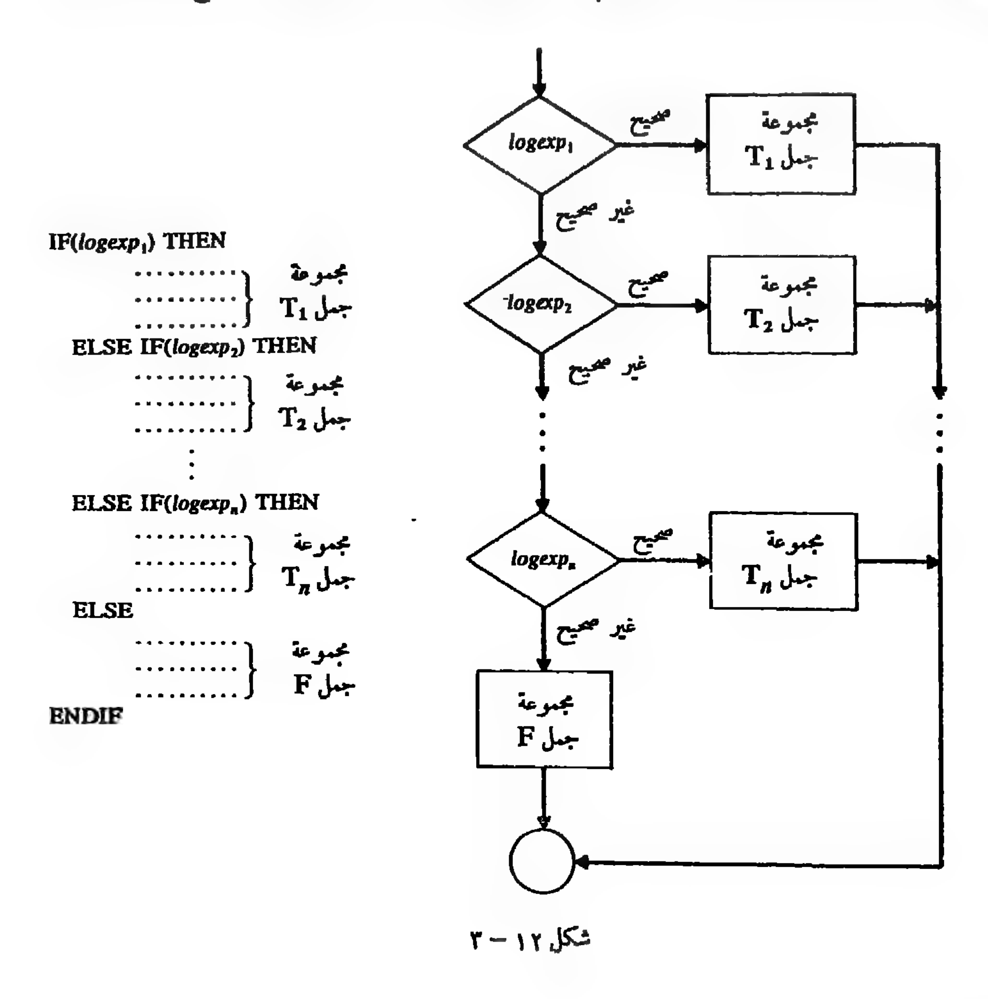
(أ)

```
IF(HOUR.LE.40.0) THEN
      GROSS = HOUR*RATE
  ELSE
      GROSS = 40.0*RATE + (HOUR – 40.0)*1.5*RATE
ENDIF
```

(ب)

```
      IF(HOUR.GT.40.0) GO TO 10
         GROSS = HOUR*RATE
      GO TO 99
10       GROSS = 40.0*RATE + (HOUR - 40.0)*1.5*RATE
99    ............................
```

ملاحظة : كما سبق أن ذكرنا فلا يسمح بنقل التحكم إلى داخل كتلة IF أو كتلة ELSE من خارج الكتلة نفسها .



شكل ١٢ – ٣

البـدائل المتعـددة

يمكننا أيضاً إضافة عدد من جمل ELSE IF بداخل هيكل IF— الأساسى لتنفيذ قرارات محددة تشمل أكثر من بديلين . يوضح شكل ١٢ – ٣ مثل هذا الهيكل ذى البدائل المتعدده مع خريطة سير العمليات الخاصة به .

وكما هو موضح فى خريطة سير العمليات ، تحسب قيمة التعبيرات المنطقية $logexp_1$ ، $logexp_2$ ، ... ... واحداً بعد الآخر . إذا كان أول تعبير منطقى $logexp_k$ صحيحاً ، فسوف تنفذ مجموعة الجمل T_k . أما إذا كانت كل التعبيرات غير صحيحة ، فسوف تنفذ مجموعة الجمل F . على أى حال ، بعد تنفذ احدى المجموعات ، فسوف ينتقل التحكم إلى أول أمر يلى جملة ENDIF.

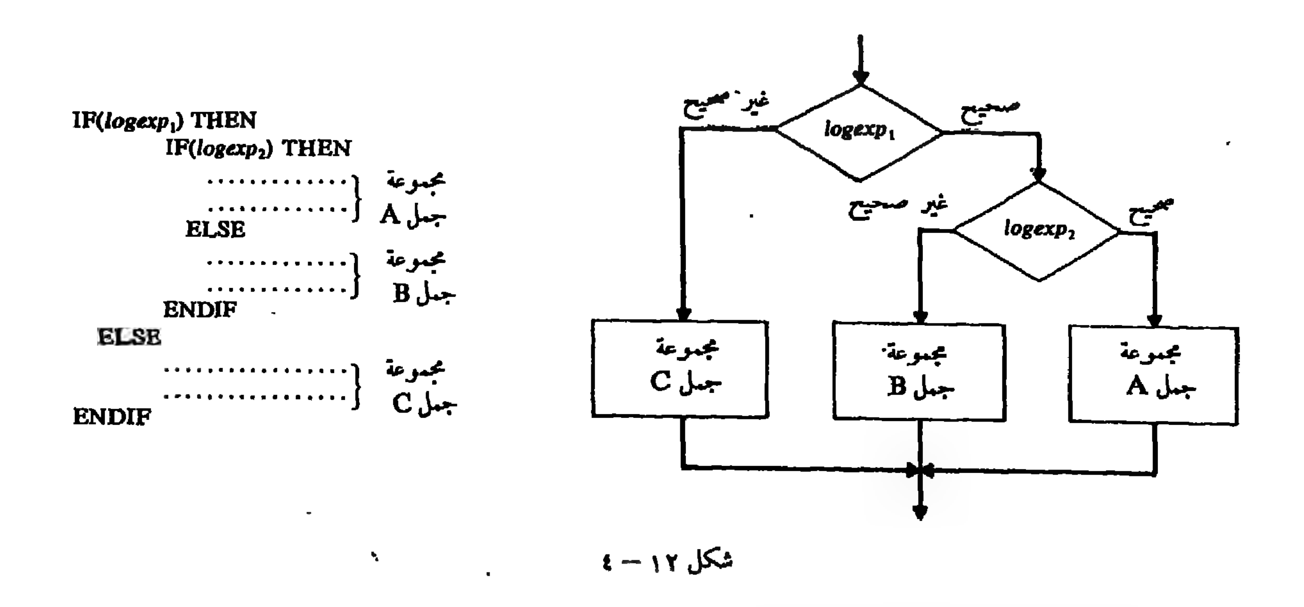
استخدام ELSE اختيارى . فمثلا تحذف الكلمة الدالة ELSE إن لم تكن هناك جمل F .

مثال ١٢ — ٣

فيما يلى جزء من برنامج الفورتران الهيكلى الذى يحسب BONUS لمندوب مبيعات . إذا كتب البرنامج بالفورتران غير الهيكل فسوف تتطلب جمل GO TO كثيرة .

```
IF(SALES.LE.200.0) THEN
        BONUS = 0.0
  ELSE IF(SALES.LE.1000.0) THEN
        BONUS = SALES*0.05
  ELSE
        BONUS = 50.0 + SALES*0.02
ENDIF
```

أخيراً ، وكما فى حلقات DO فلا يمكن لهياكل IF— أن تتداخل . ومع ذلك فيمكن أن يحتوى هيكل IF— على مجموعة كاملة من جمل هيكل IF— أخرى . ويوضح ذلك هيكل البرنامج وخريطة سير العمليات الخاصة به فى شكل ١٢ — ٤ .



شكل ١٢ — ٤

١٢ — ٢ هياكل التحكم فى الحلقة التكرارية

سنناقش ثلاثة هياكل للتحكم فى الحلقة التكرارية : حلقة WHILE وحلقة FOR وحلقة DO العامة . تنفذ الحلقتان الأوليتان على كثير من المشغلات ، بما فيها مشغل WATFIV—S والحلقة الثالثة متضمنة فى بعض نسخ من فورتران 77 .

حلقة DO الخاصة بالفورتران غير الهيكلى والتى نوقشت فى الفصل الخامس هيكل مفيد ومناسب للتحكم فى الحلقة التكرارية . ومع ذلك فهناك عدة قيود غير مناسبة ، فمثلا يجب أن يكون المتغير الذى يتحكم فى الحلقة التكرارية (أى الدليل) متغير من النوع الصحيح وكذلك لايمكن أن تكون معاملات الحلقة التكرارية تعبيرات حسابية ، وليس فى استطاعتنا أيضاً أن ننقص من قيمة الدليل ..هذه العوائق ادت إلى إنشاء عدة هياكل لحلقات تكرارية أخرى . ومن الهياكل الأكثر شيوعاً هيكل WHILE— وهيكل FOR — التى سوف نناقش فيما بعد

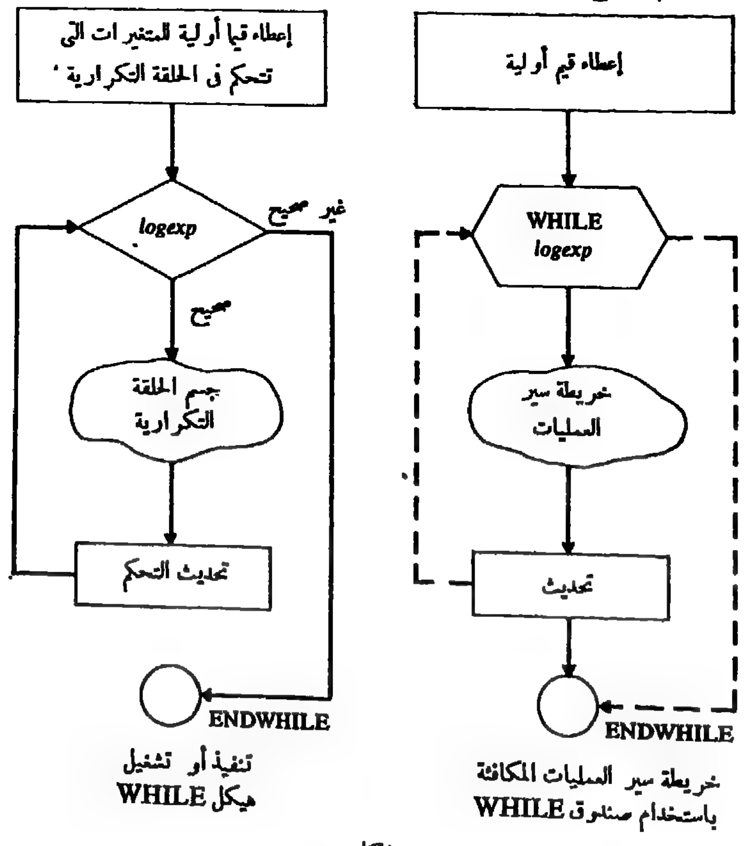
هيكل — WHILE

لايمكن أن نتحكم بسهولة فى كثير من العمليات التكرارية بواسطة عداد . فى مثل هذه الحالات يمكن استخدام هيكل —WHILE بطريقة ملائمة . فيما يلى الشكل العام لهذا الهيكل .

```
WHILE(logexp) DO
        ..............  }
        ..............  }  جسم الحلقة التكرارية
        ..............  }
ENDWHILE
```

تحسب أولا قيمة التعبير المنطق *logexp* ، فإذا كان صحيحاً ، فسوف ينفذ جسم الحلقة التكرارية . وعلاوة على ذلك سيتكرر جسم الحلقة التكرارية ما دام التعبير المنطق *logexp* صحيحاً – وسوف تتخطى جسم الحلقة التكرارية عندما يكون التعبير المنطق *logexp* غير صحيح وينتقل التحكم إلى أول جملة بعد جملة ENDWHILE . لاحظ أن جملتى WHILE و ENDWHILE تستخدمان كحدود تحيطان بالهيكل التكرارى . وأنه لغاية فى الأهمية أن نتذكر أننا دائماً نحسب قيمة التعبير المنطق أولا قبل تنفيذ الحلقة التكرارية .

عادة يمكن أن يشتمل التعبير المنطق *logexp* فى هيكل —WHILE على متغير واحد أو أكثر . يجب أن تعطى هذه المتغيرات ، (وتسمى المتغيرات التى تتحكم فى الحلقة التكرارية) ، قيما أولية قبل البدء فى تنفيذ هيكل —WHILE . ولتجنب وجود حلقة تكرارية لا نهائية ، يجب أيضاً أن تعدل ، أو تحدث قيم المتغيرات مع كل تكرار لحلقة التكرارية ، ويتم التحديث أو التعديل عادة قبل جملة ENDWHILE . ولقد تم توضيح ذلك فى شكل ١٢ – ٥ كخريطة سير العمليات التى تمثل هيكل WHILE .



شكل ١٢ – ٥

مثال ١٢ - ٤

أدرس كثيرة الحدود من الدرجة الثانية $y = 2x^2 - 3x - 5$ يحسب برنامج الفورتران التالى الذى يستخدم هيكل - WHILE قيمة y لقيم x التى تتراوح مابين 4 - إلى 4 بخطوات 0.5 .

```
      X = -4.0
      WHILE(X.LE.4.0) DO
            Y = 2.0*X**2 - 3.0*X - 5.0
            WRITE(6, 10) X, Y
10          FORMAT(2(3X, F10.3))
            X = X + 0.5
      ENDWHILE
```

وقارن ببساطة البرنامج السابق بالبرنامج فى المسألة ه - ١١ (أ) ، الذى يقوم بعمل نفس الشىء ولكن باستخدام حلقة DO الأساسية ذات الدليل . كان يجب فى الحالة السابقة أن تعد مرات التكرارات وأيضاً تكتب X بدلالة الدليل I .

هيكل —FOR

وكما ذكرنا سابقاً فإن أحد القيود الصارمة لحلقة DO غير الهيكلية هو عدم مقدرتها على إنقاص قيمة الدليل . عائق آخر وهو عدم اختيار الدليل لقيمة النهاية إلا بعد أول تكرار . وعلى ذلك فرغم أن M > N فإن حلقة DO الأساسية (غير الهيكلية) المبتدئة بجملة DO التالية :

```
DO 50 K = M, N
```

ستنفذ مرة واحدة .

لقد تخلصنا من هذه العوائق فى هيكل —FOR والتى لها الشكل التالى :

```
FOR(INDEX = INV, ENDV, INCR) DO
      ..............................
      ..............................   } جسم الحلقة التكرارية        (أ)
      ..............................
ENDFOR
```

الدليل INDEX يجب أن يكون إسماً متغيراً صحيحاً . ومع ذلك فإن كلا من INV و ENDV و INCR والتى تمثل القيمة الابتدائية وقيمة النهاية ومعامل بزيادة على الترتيب ، يمكن أن تكون أى تعبير صحيح فيما عدا أن معامل الزيادة INCR لايمكن أن يأخذ القيمة صفراً .

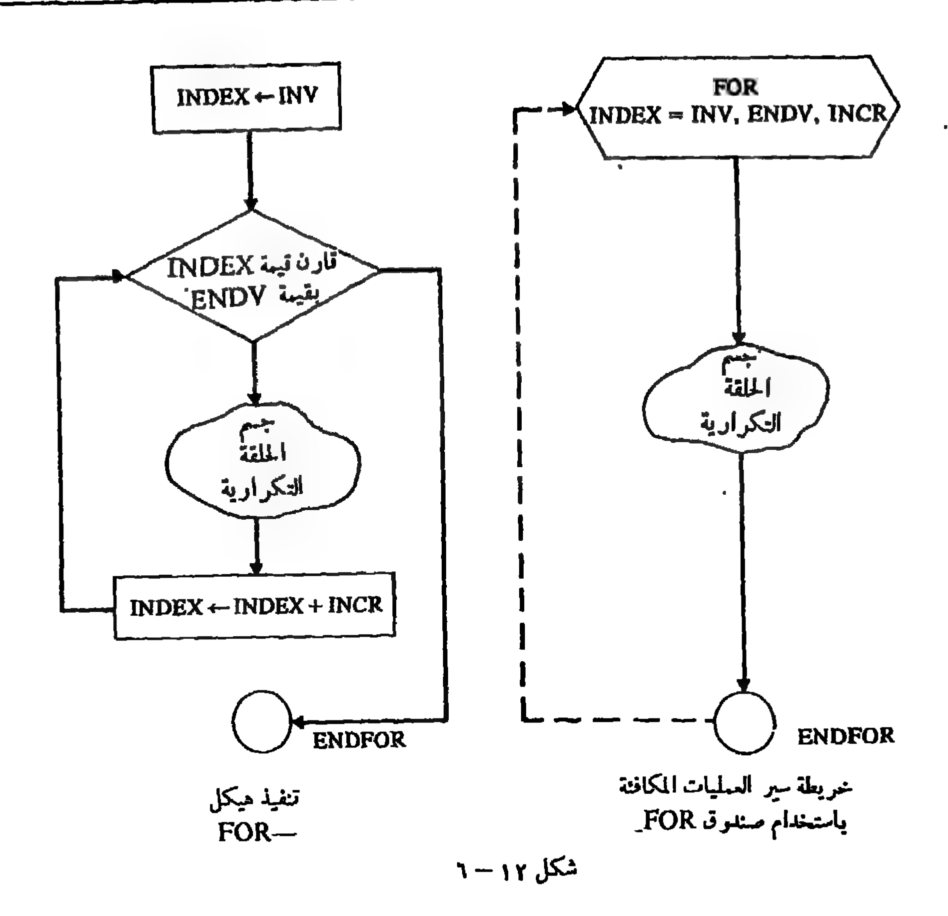
يوضح شكل ١٢ - ٦ خريطة سير العمليات لتنفيذ هيكل —FOR ، تجعل قيمة الدليل INDEX فى البداية مساو INV . تختبر قيمة INDEX قبل تنفيذ الحلقة التكرارية ، لنرى ما إذا تعدت قيمة النهاية المعطاة ENDV . يختلف هذا الاختبار حسب قيمة INCR هل موجبة أم سالبة ؟ إذا كانت INCR موجبة ، يكون السؤال :

is INDEX ≤ ENDV ?

ومع ذلك فإذا كانت INCR سالبة ، يكون السؤال :

is INDEX ≥ ENDV ?

نستمر فى تنفيذ الحلقة التكرارية طالما لم تتعد قيمة INDEX القيمة ENDV أى طالما تكون نتيجة الاختبار « نعم » .



تنفيذ هيكل
FOR—

خريطة سير العمليات المكافئة
باستخدام صندوق FOR

شكل ١٢ — ٦

تزاد قيمة INDEX بمقدار INCR بعد تنفيذ الحلقة التكرارية وقبل التكرار التالى .

حيث أن معامل الزيادة INCR يمكن أن يكون سالباً ، لذلك (ممكن إنقاص قيمة INDEX) والمثال التالى يوضح هذه الميزة :

مثال ١٢ — ٥

إفرض أننا نريد طباعة الأعداد الصحيحة 100 و 99 و 98 و . . . . و 1 وبهذا الترتيب يمكننا عمل ذلك بطريقتين : (١) بحلقة — DO الأساسية (غير الهيكلية) (أنظر مثال ٥ — ٥) و (٢) بهيكل — FOR .

(أ)

```
      FOR(K = 100, 1, -1) DO
          WRITE(6, 30) K
30        FORMAT(1X, I5)
      ENDFOR
```

(ب)

```
      DO 500 K = 1, 100
          J = 101 - K
          WRITE(6, 30) J
30        FORMAT(1X, 5I)
500   CONTINUE
```

لاحظ أن الحلقة التكرارية فى (ب) تنتهى عندما تأخذ K القيمة 0 أى بعد أن تتجاوز قيمة النهاية 1 .

حلقة DO المعممة (حلقة — DO الهيكلية)

تتجاهل هياكل —WHILE وهياكل —FOR بعض العوائق الموجودة فى حلقة —DO الأساسية (غير الهيكلية) التى تمت مناقشتها فى الفصل الخامس . يحتوى فورتران 77 الجديد بناء حلقة —DO المعممة التى تجمع خصائص كل من هيكل —WHILE وهيكل —FOR فى بناء واحد .

والحلقة –DO المعممة هذه نفس المظهر الأساسى لحلقة –DO ذات الدليل التى تم عرضها فى الأقسام ٥ – ٣ و ٥ – ٤ وسيكون لها الشكل التالى :

```
      DO n VAR = INV, ENDV, INCR
          ........................
          ..........  ..........   }  DO – جسم حلقة
          ........................
    n   CONTINUE
```

مرة ثانية تحيط بجسم الحلقة التكرارية المراد تكراره بزوج الجمل DO-CONTINUE .

يسمى VAR متغير التحكم فى الحلقة ، ويمكن أن يكون اسم متغير صحيح أو اسم متغير حقيقى ، تمثل معاملات الحلقة التكرارية INV و ENDV و INCR القيمة الابتدائية وقيمة النهاية (أو قيمة الاختبار) ومعامل الزيادة على الترتيب ، ويمكن أن تكون أى تعبيرات صحيحة أو حقيقية . (وهذا يتعارض مع حلقة – DO المشروحة فى الفصل الخامس حيث يجب أن يكون الدليل متغيراً صحيحاً وأن تكون المعاملات ثوابت صحيحة موجبة أو متغيرات صحيحة) . وإن لم يكن INCR موجوداً ، فتعتبر معامل الزيادة 1 ، ولذلك ، لايمكن أن تكون INCR له قيمة صفرا .

من المناقشة فى الفصل الخامس وتبعاً لهيكل –FOR السابق فإن الصيغة DO=CONTINUE السابقة تقترح أن نبدأ التكرار بجعل قيمة VAR الابتدائية مساويه لقيمة INV ، ثم تزداد قيمة VAR بمقدار INCR مع كل تكرار ، ولستمر فى تكرار الحلقة حتى تتجاوز قيمة VAR قيمة ENDV . هذا التفسير صحيحاً أساساً ، ومع ذلك فن الضرورى أعطاء معنى دقيق لهذا البناء العام أى لحلقة –DO المعممة .

تظهر خريطة سير العمليات لتنفيذ حلقة –DO المعممة فى شكل ١٢–٧ ، عند تنفيذ جملة –DO تحسب قيم INV وENDV وINCR ثم تخصص القيمة الابتدائية INV للمتغير VAR وتحسب قيمة عداد التكرار والذى رمزنا له بالرمز KOUNT . وعادة يمكن أن تحسب عداد التكرار KOUNT كنتيجة للقيمة الصحيحة (ENDV — INV + INCR)/INCR فمثلا :

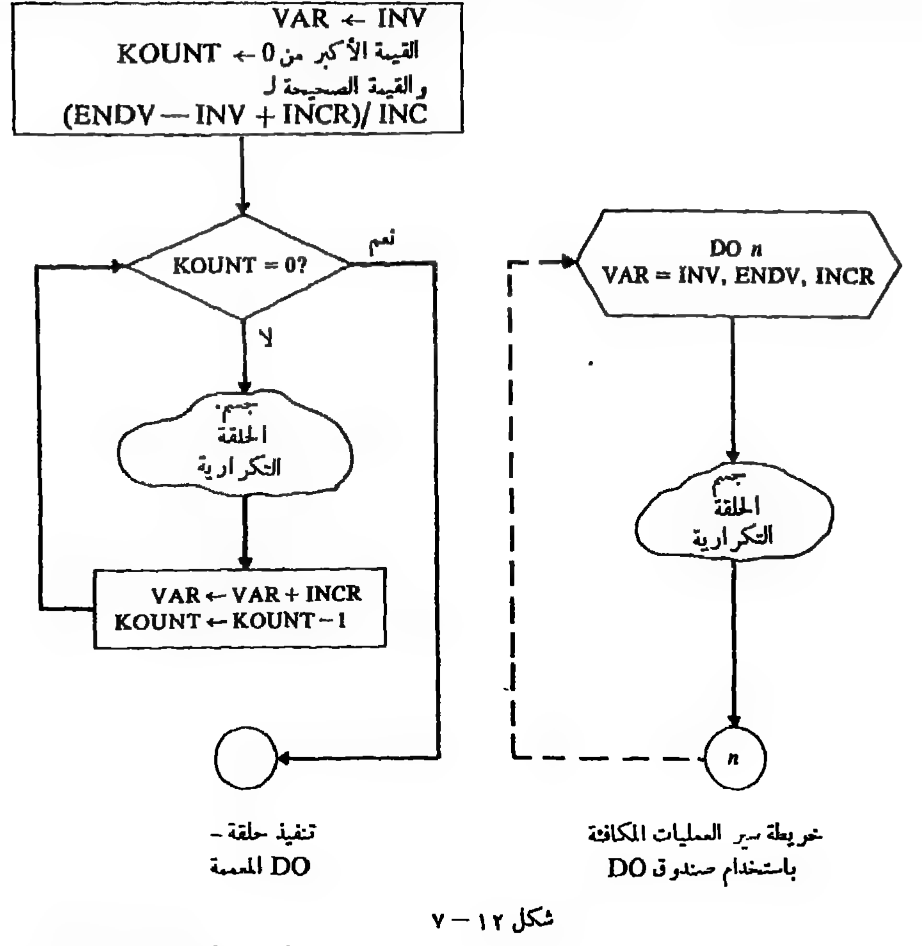
KOUNT = 4 تعطى DO 100 K = 2, 13, 3
KOUNT = 21 تعطى DO 200 K = 2.0, 1.0, −0.05

لعمل حساب المواقف التى يكون فيها INV > ENDV و INCR موجبة ، أو عندما تكون INV<ENDV و INCR سالبة (أى ، عندما تتجاوز قيمة VAR من البداية قيمة النهاية ENDV) ، تخصص إلى KOUNT القيمة الأكبر من الصفر والقيمة الصحيحة لـ (ENDV — INV + INCR)/INCR) . ونؤكد أن هذا العداد KOUNT يعطى عدد المرات التى ستنفذ فيها الحلقة التكرارية .

وكما هو موضح بالشكل ١٢ – ٧ نختبر ما إذا كانت قيمة KOUNT تساوى صفر قبل تنفيذ الحلقة التكرارية . إذا كانت KOUNT ≠ 0 فسوف تنفذ الحلقة التكرارية وتزداد قيمة VAR بمقدار INCR وتنقص عداد التكرار KOUNT بمقدار واحد . وينتقل التحكم إلى أول جملة منفذة تلى جملة CONTINUE عندما تصبح قيمة KOUNT = 0 .

مثال ١٢ – ٦

(أ) إفرض أننا نريد أن نطبع الأعداد الصحيحة 100 و 99 و ... و 1 بهذا الترتيب . فباستخدام حلقة –DO المعممة يكون لدينا مايلى :



تنفيذ حلقة DO المعممة

خريطة سير العمليات المكافئة باستخدام صندوق DO

شكل ١٢ — ٧

```
       DO 500 K = 100, 1, −1
          WRITE(6, 30) K
30        FORMAT(1X, I5)
500    CONTINUE
```

(ب) وباستخدام حلقة DO المعممة لحل المسألة في مثال ١٢ — ٤ يكون لدينا مايلي :

```
       DO 999 X = −4.0, 4.0, 0.5
          Y = 2.0*X**2 − 3.0*X − 5.0
          WRITE(6, 10) X, Y
10        FORMAT(2(3X, F10.3))
999    CONTINUE
```

يمكن تلخيص خصائص DO المعممة كالآتي :

١ — كما هو معتاد إذا لم توجد INCR فنعتبر مقدار الزيادة 1 .

٢ — ليس من الضروري أن تكون أنواع قيم معاملات الحلقة التكرارية ENDV و INCR و INV من نفس نوع المتغير VAR . في مثل هذه الحالة تحول أولا قيم المعاملات إلى نوع المتغير VAR .

٣ — يمكن أن تكون قيمة INCR سالبة وعلى ذلك من الممكن أن تنقص قيمة VAR . على أي حال، فإننا نستمر في تنفيذ الحلقة التكرارية لكل قيم VAR إلى أن تتجاوز قيمة VAR قيمة النهاية ENDV .

٤ – لايمكن تغيير قيمة متغير الحلقة التكرارية VAR بداخل جسم حلقة DO إلا بطريقة الزيادة الخاصة به .

٥ – يعرف دائماً متغير الحلقة التكرارية عندما ينتقل التحكم من حلقة –DO ، (يتعارض هذا مع دليل حلقة DO الأساسية (غير الهيكلية) ، وفيها يعرف فقط فى حالة الخروج غير الطبيعى من الحلقة التكرارية) .

٦ – إذا كانت قيمة INV > ENDV و INCR موجبة ، أو إذا كانت INV < ENDV و INCR سالبة سيكون العداد KOUNT قيمة إبتدائية صفر ولن ينفذ جسم الحلقة التكرارية على الإطلاق .

٧ – لايسمح بنقل التحكم إلى وسط حلقة DO المعممة من الخارج (كما فى حلقة –DO الأساسية ذات الدليل) . تطبق هنا أيضاً قواعد حلقات DO المتداخلة التى تم مناقشتها فى قسم ٥ – ٩ .

٨ – لايمكن أن تتداخل حلقات DO وهياكل –IF. ومع ذلك ، فيمكن أن تحتوى كتلة جملة هيكل IF على حلقة DO. ويمكن أن تحتوى حلقة DO على هيكل –IF .

ملاحظة : إذا كان متغير الحلقة التكرارية VAR متغيراً صحيحاً ، فإن تنفيذ حلقة DO التكرارية المعممة يشبه هيكل –FOR التى تمت مناقشته سابقاً . وفى الحقيقة تتصرف حلقة DO المعممة بطريقة مماثلة جداً لحلقة DO الأساسية ذات الدليل فيما عدا : (أ) أنه يمكن أن تكون INCR سالبة ، (ب) ويمكن أن تكون معاملات الحلقة التكرارية تعبيرات (جـ) وأن الاختبار يتم قبل تنفيذ الحلقة التكرارية .

مســائل محلولة

هياكل –IF

١٢ – ١ إفرض أن نسبة RATE الفائدة على قرض AMT هى 7% إذا كان القرض AMT ≤ $ 10,000,00 . وتكون النسبة 6% إذا تعدت AMT مبلغ $ 10,000.00 بفرض أن INT متغير حقيقى وأن AMT مخزنة فى الذاكرة ، اكتب البرنامج الذى يحدد قيمة INT على القرض .

يحتوى شكل ١٢ – ٨ مثل هذا البرنامج (قارنه مع البرنامج غير الهيكلى على صفحة ١٠١)

```
      IF(AMT.LE.10000.0) THEN
            RATE = 0.07
        ELSE
            RATE = 0.06
      ENDIF
      INT = AMT*RATE
      WRITE(6, 10) AMT, RATE, INT
   10 FORMAT(3(5X, F10.2))
```

شكل ١٢ – ٨

```
      IF(TYPE.EQ.1) THEN
            NET = PAY - 9.75
        ELSE IF(TYPE.EQ.2) THEN
            NET = PAY - 16.25
        ELSE IF(TYPE.EQ.3) THEN
            NET = PAY - 24.50
      ENDIF
      WRITE(6, 20) ID, NET
   20 FORMAT(1X, I5, 3X, F12.2)
```

شكل ١٢ – ٩

١٢ – ٢ إفرض أن قسط التأمين الصحى يخصم من مرتب موظف تبعاً للخطة التالية (حيث TYPE متغير صحيح) :

القسط =
- 9.75 إذا كانت TYPE = 1 (أعزب)
- 16.25 إذا كانت TYPE = 2 (متزوج ولا يعول)
- 24.50 إذا كانت TYPE = 3 (متزوج ويعول)

اكتب جزء برنامج يخصم القسط من مرتب موظف بفرض أن PAY و TYPE و ID مخزنة فى الذاكرة (وأن NET متغير حقيقى) .

تظهر خريطة سير العمليات لمثل هذا البرنامج وترجمته إلى الفورتران غير الهيكلى على صفحة ١٠٨. يحتوى شكل ١٢ – ٩ ترجمة الفورتران الهيكلى باستخدام هيكل -IF ذى البدائل المتعددة .

١٢ – ٣ اكتب جزء البرنامج الذى يقرأ عدد صحيحاً M ويطبع ما إذا كان هذا العدد موجباً أو سالباً أو صفر .

يحتوى شكل ١٢ – ١٠ على هذا البرنامج باستخدام هيكل -IF ذى البدائل المتعددة .

```
      READ(5, 10) M
10    FORMAT(I5)
      WRITE(6, 20) M
20    FORMAT(1X, I5, 2X, 'IS')
      IF(M.GT.0) THEN
            WRITE(6, 30)
30          FORMAT(1X, 'POSITIVE')
       ELSE IF(M.EQ.0) THEN
            WRITE(6, 40)
40          FORMAT(1X, 'ZERO')
       ELSE
            WRITE(6, 50)
50          FORMAT(1X, 'NEGATIVE')
      ENDIF
```

شكل ١٢ – ١٠

```
      READ(5, 10) A, B, C
10    FORMAT(3F10.2)
      WRITE(6, 20) A, B, C
20    FORMAT(1X, 'SIDES', 3(2X, F10.2))
      IF(A.EQ.B.AND.B.EQ.C) THEN
            WRITE(6, 30)
30          FORMAT(1X, 'EQUILATERAL')
       ELSE IF(A.EQ.B.OR.A.EQ.C.OR.B.EQ.C) THEN
            WRITE(6, 40)
40          FORMAT(1X, 'ISOSCELES')
      ENDIF
```

شكل ١٢ – ١١

١٢ – ٤ اكتب جزء البرنامج الذى يقرأ الأطوال A و B و C لمثلث T ويطبع ما إذا كان T مثلث متساوى الأضلاع أو مثلث متساوى الساقين .

يكون T مثلث متساوى الأضلاع إذا كانت A = B و B = C . ويكون T مثلث متساوى الساقين إذا كانت A = B أو A = C أو B = C . يحتوى شكل ١٢ – ١١ على مثل هذا البرنامج (نستخدم الروابط المنطقية AND. و OR. التى تم مناقشتها فى الفصل التاسع) .

هياكل -DO.

١٢ – ٥ أوجد عداد التكرار KOUNT لكل جمل DO التالية :

(أ) DO 100 K = 4, 20, 3 (جـ) DO 300 K = 4, −9, −2
(ب) DO 200 X = −4, 4, 0.6 (د) DO 400 X = 5, −2, 0.2

أولا أوجد خارج القسمة $Q = (ENDV - INV + INCR)/INCR$ إجعل J ترمز إلى القيمة الصحيحة Q . تكون بعد ذلك KOUNT مساوية J أو 0 أيهما أكبر .

(أ) $Q = (20 - 4 + 3)/3 = 19/3$ ؛ إذن J = 6 . من ثم KOUNT = 6 .

(ب) $Q = (4 + 4 + 0.6)/0.6 = 8.6/0.6$ ؛ إذن J = 14 . من ثم KOUNT = 14 .

(جـ) $Q = (-9 - 4 - 2)/2 = -15/-2$ ؛ إذن J = 7 . من ثم KOUNT = 7 .

(د) $Q = (-2 - 5 + 0.2)/0.2 = -6.8/0.2$ ؛ إذن J = −34 . من ثم KOUNT = 0 .

١٢ – ٦ أوجد القيمة النهائية لكل من K و L .

(أ)
```
     DO 100 K = 1, 5
        L = K**2
100  CONTINUE
     K = 2*K
```

(ب)
```
     DO 200 K = 5, 1
        L = K**2
200  CONTINUE
     K = 2*K
```

(أ) تنفذ حلقة DO لقيم K = 1, 2, 3, 4 ثم للقيمة 5 من ثم تكون القيمة النهائية لـ L هى $5^2 = 25$. ومع ذلك فعندما ينتقل التحكم من الحلقة التكرارية ، فتزاد قيمة K وتصبح 6 . من ثم تكون قيمة K النهائية 12 .

(ب) لاتنفذ حلقة DO على الإطلاق حيث أن عداد التكرار KOUNT = 0 ؛ من ثم تكون L غير معرفة . ومع ذلك ، تخصص القيمة الابتدائية 5 إلى K عند تنفيذ جملة DO . من ثم تكون قيمة K النهائية 10 .

١٢ – ٧ إفرض أن $z = x^3 - y^2$ اكتب جزء البرنامج الذى يوجد z لقيم x و y حيث تتغير كل من x و y من 4 — إلى 4 بخطوات مقدارها 0.5 .

يحتوى شكل ١٢ – ١٢ على مثل هذا البرنامج الذى يستخدم حلقات DO المعممة المتداخلة (قارن هذا البرنامج بالبرنامج المشابه فى المسألة ٥ – ١١ (ب)) .

```
     DO 200 X = -4, 4, 0.5
        DO 100 Y = -4, 4, 0.5
           Z = X**3 - Y**2
           WRITE(6, 10) X, Y, Z
10         FORMAT(3(2X, F10.2))
100     CONTINUE
200  CONTINUE
```

شكل ١٢ – ١٢

```
FOR(J = N, K, -1) DO
     A(J + 1) = A(J)
ENDFOR
A(K) = D
```

شكل ١٢ – ١٣

١٢ – ٨ اكتب جزء البرنامج باستخدام هيكل-FOR الذى يضيف عنصر D فى المكان K من مجموعة متراصة A(1) ، A(2)....A(N)

كما تمت مناقشته فى مسألة ٦ – ١٠ ، يجب أن نحرك جزء المجموعة المتراصة A(K) و....وA(N) إلى أسفل مكان واحد قبل تخصيص D إلى A(K) . يتم هذا بتخصيص A(N) إلى A(N + 1) أولا ، ثم A(N — 1) إلى A(N) وهكذا إلى أن نصل A(K) إلى A(K + 1) يحتوى شكل ١٢ – ١٣ على مثل هذا البرنامج .

١٢ – ٩ اكتب برنامجاً باستخدام هيكل-WHILE الذى يطبع كل الأعداد الصحيحة الموجبة المفردة أقل من 100 مع حذف مضاعفات 7 :
1, 3, 5, 9, 11, 13, 15, 17, 19, 23, ... 97, 99

يحتوى شكل ١٢ – ١٤ على مثل هذا البرنامج . لاحظ كيف تعطى N قيمة إبتدائية قبل الحلقة التكرارية وكيف يتم تحديثها (تعديلها) عند نهاية الحلقة التكرارية .

```
     N = 1
     WHILE(N.LT.100) DO
          K = (N*7)/7
          IF(K.EQ.N) GO TO 5
          WRITE(6, 10)
10        FORMAT(10X, I2)
5         N = N + 2
     ENDWHILE
```

شكل ١٢ – ١٤

```
     DIMENSION SCORE(300)
     N = 1
     READ(5, 10) GRADE
10   FORMAT(F8.0)
     WHILE(GRADE.GE.0.0) DO
          SCORE(N) = GRADE
          N = N + 1
          READ(5, 10)GRADE
     ENDWHILE
     N = N - 1
```

شكل ١٢ – ١٥

١٢ – ١٠ تم تثقيب درجات اختبار لطلبة درجة واحدة على كل بطاقة ، والمجموعة لها بطاقة خلفية مثقب عليها رقم سالب . بفرض وجود أقل من 300 درجة ، أكتب جزء برنامج باستخدام هيكل -WHILE الذى يحسب عدد الدرجات N ويخزنها فى مجموعة متراصة SCORE .

يحتوى شكل ١٢ – ١٥ على مثل هذا البرنامج . لاحظ أن N تنقص بمقدار واحد بعد نقل التحكم من الحلقة التكرارية (لماذا ؟) .

مســائل تكميلية

هياكل -IF و -DO

١٢ – ١١ إفرض أن J = 5 و K = 10 . أوجد القيمة النهائية لكل من J و K إذا نفذنا ما يلى :

(أ)
```
IF(J.LE.10) THEN
      K = K + 5
      J = J + K
ENDIF
J = J + K
J = 2*J
```

(ب)
```
IF(J.LE.10) THEN
      K = K + 5
      J = J + K
  ELSE
      J = J + K
ENDIF
J = 2*J
```

(حـ)
```
IF(J.GT.10) THEN
      K = K + 5
      J = J + K
ENDIF
J = J + K
J = 2*J
```

(د)
```
IF(J.GT.10) THEN
      K = K + 5
      J = J + K
  ELSE
      J = J + K
ENDIF
J = 2*J
```

١٢ – ١٢ أدرس أجزاء البرامج التالية :

(١)
```
M = 5
IF(J.LT.K) THEN
      J = J + 5
      M = M + 3
IF(J.LT.8) THEN
      M = M + 10
  ELSE
      M = M + 20
ENDIF
```

(٢)
```
M = 5
IF(J.LT.K) THEN
      J = J + 5
      M = M + 3
  ELSE IF(J.LT.8) THEN
      M = M + 10
  ELSE
      M = M + 20
ENDIF
```

أوجد قيمة M النهائية لكل برنامج إذا كانت :

(أ) J = 2, K = 5 (ب) J = 2, K = 1 (حـ) J = 10, K = 15 (د) J = 10, K = 1

١٢ – ١٣ أوجد عداد التكرار KOUNT لكل جملة من جمل DO التالية :-

(أ) DO 100 K = −3, 19, 4
(ب) DO 200 X = 4, −6, −0.3
(حـ) DO 300 K = 2, 25, −6
(د) DO 400 X = 1, −3, −0.2

١٢ – ١٤ أوجد القيمة النهائية لكل من J و M إذا نفذنا ما يلى :

(أ)
```
    DO 100 J = 2, 9, 3
       M = J + 5
100 CONTINUE
```

(ب)
```
    DO 200 J = 9, 2, -5
       M = J + 5
200 CONTINUE
```

(حـ)
```
    DO 300 J = 9, 2, 3
       M = J + 5
300 CONTINUE
```

(د)
```
    DO 400 J = 2, 3; 9
       M = J + 5
400 CONTINUE
```

١٢ – ١٥ أوجد القيمة النهائية لكل من J و K و L بعد تنفيذ مايلى :

(أ)
```
    DO 100 J = 2, 6. 3
       DO 200 K = J, 4
          L = J + K
200    CONTINUE
100 CONTINUE
```

(ب)
```
J = 3
WHILE(J.LT.6) DO
   K = 2*J
   L = 3*J
   J = J + 2
ENDWHILE
```

برامج :

١٢ – ١٦ يخصص لدرجة الاختبار T الحرف A أو B أو C أو D أو F تبعاً لما يأتى :

$T \geq 90$ أو $80 \leq T < 90$ أو $70 \leq T < 80$ أو $60 \leq T < 70$ أو $T < 60$

اكتب البرنامج الذى يقرأ درجة الاختبار ويطبع الحرف المقابل له .

١٢ – ١٧ تحتوى كل بطاقة فى المجموعة درجة اختبار الطالب ، والمجموعة لها بطاقة (خلفية) مثقب بها رقم سالب . اكتب البرنامج الذى يجد عدد الطلبة N الذين أدوا الاختبار وعدد الطلبة M الذين كانت أوراقهم ممتازة (أى ، أخذوا 100) ، وعدد الطلبة L الذين رسبوا (أى حصلوا على < 60) .

١٢ – ١٨ اكتب البرنامج الذى يحسب الجذور الحقيقية لمعادلة الدرجة الثانية :

$$ax^2 + bx + c = 0$$

مع معلومية المعاملات A و B و C حيث $A \neq 0$ (قارن مع البرنامج على صفحة ١٠٩) .

١٢ – ١٩ اكتب البرنامج الذى يقرأ ثلاثة أرقام موجبة A و B و C ويحدد ما إذا كانت A و B و C أطوال أضلاع مثلث . وإذا كانت الإجابة نعم احسب محيط المثلث ، أما إذا كانت الإجابة لا اطبع الرسالة « NOT A TRIANGLE » (ليس مثلثاً) . (تلميح : لايمكن أن تشكل A و B و C مثلثاً إلا إذا كان طول ضلع ما أكبر من أو يساوى مجموع الضلعين الآخرين) . اختبر البرنامج بالبيانات :

(أ) A = 3.0, B = 4.0, C = 5.0 (ب) A = 3.0, B = 12.0, C = 5.0

١٢ – ٢٠ اكتب برنامجاً باستخدام حلقة DO معممة يضيف عنصر D فى المكان K فى مجموعة مراصة A(1) و A(2) و . . . و A(*n*) (قارن مع المسائل ١٢ – ٨ و ١٢ – ١٠)

١٢ - ٢١ أوجد عدد النقط ذات الاحداثيات الصحيحة والتى تقع داخل :

(أ) الدائرة $x^2 + y^2 = 50$ (ب) القطع الناقص $2x^2 + 3y^2 = 100$ (قارن مع المسألة ٥ - ١٢) .

١٢ - ٢٢ اكتب البرنامج الذى يطبع الرقم 20 عشرين مرة والرقم 19 تسع عشرة مرة والرقم 18 ثمانى عشرة مرة ، وهكذا .

١٢ - ٢٣ تحتوى كل بطاقة فى مجموعة بطاقات على عدد صحيح موجب . والمجموعة لها بطاقة خلفية خالية . أكتب البرنامج الذى يحدد عدد الأرقام الصحيحة الزوجية وعدد الأرقام الصحيحة الفردية .

١٢ - ٢٤ إدرس المعادلة $y = x^2 - 4x + b$. اكتب البرنامج الذى يحدد y لقيم x إبتداء من I إلى J بزيادة قدرها D ، وحيث I و J و D مثقبة على بطاقة و $I < J$ و $D > 0$.

١٢ - ٢٥ إفرض أنك أودعت مبلغ \$2,000.00 فى حساب توفير يعطى 7% فائدة مركبة سنوياً . اكتب برنامجاً يستخدم هيكل -WHILE لتحديد عدد السنوات التى تنقضى ليصل الحساب إلى \$5,000.00 .

١٢ - ٢٦ قررت إدارة محل أن تعطى خصماً على ثمن PRICE التليفزيون الملون تبعاً لحجم الجهاز SIZE بالبوصة كما يلى :

الخصم =
- \$40,00 إذا كان الحجم 12 أو 15
- \$70.00 إذا كان الحجم 17 أو 19
- \$110.00 إذا كان الحجم 21 أو 23
- \$140.00 إذا كان الحجم 25

هناك خصم إضافى مقداره 3% من الثمن PRICE إذا كان $300 \leq PRICE < 400$ و 4% إذا كان $PRICE > 400$. اكتب البرنامج الذى يقرأ الرقم ID لكل TV وكذا حجمه SIZE وثمنه PRICE ويحدد ثمن TV الصافى NET (إفرض أن SIZE متغير صحيح و NET متغير حقيقى)

١٢ - ٢٧ هناك أربعة مرشحين لمنصب فى مجلس مدرسة . يدلى الشخص بصوته بواسطة تثقيب 1 أو 2 أو 3 أو 4 على بطاقة . سيكون لمجموعة البطاقات بطاقة خلفية مثقب عليها 1 - . اكتب البرنامج الذى يحصى عدد الأشخاص N الذى أدلوا بأصواتهم ، وعدد الأصوات - التى حصل عليها كل مرشح .

١٢ - ٢٨ أعد كتابة المسألة ١٢ - ٢٧ ، غير أن البرنامج هنا يحدد أيضاً الفائز فى الانتخابات .

١٢ - ٢٩ أعد كتابة ١٢ - ٢٧ غير أن فى هذه المرة عدد المرشحين لمنصبين خمسة والبرنامج يحدد أيضاً الفائزين .

حلول المسائل المختارة

١٢ – ١١

(أ) 70, 15 (ب) 40, 15 (حـ) 30, 10 (د) 30, 10

١٢ – ١٢

(أ) (*i*) 18 (*ii*) 8 (حـ) (١) 28 (٢) 8

(ب) (*i*) 15 (*ii*) 15 (د) (١) 25 (٢) 25

١٢ – ١٣

(أ) 5 (ب) 33 (حـ) 0 (د) 20

١٢ – ١٤ (أ) J = 11, M = 13 (حـ) (ليست معرفة) J = 9, M

(ب) J = —1, M = 9 (د) J = 11, M = 7

١٢ – ١٥

(أ) J = 8, K = 5, L = 4 (ب) J = 7, K = 10, L = 15

ملحق (١)

دوال المــكتبة

يعطى الملحق قائمة لدوال المكتبة (المبيتة – أو المبنية داخلياً) والمتوافرة لمنظم نظم الفورتران . يشار إلى خلاصات الدالة فى جدول أ – ١ بالرموز التالية . والتى تدل على أنواعها .

K و J نوع صحيح	DX و DY نوع متضاعف الدقة
Y و X نوع حقيق	C نوع مركب

سنستخدم كلمة متضاعفة (*double*) للدقة المتضاعفة . يرمز إلى إشارة الخلاصة a بالرمز $\text{sgn}(a)$ وتعرف بالآتى :

تكون $1 = \text{sgn}(a)$ إذا كانت $a \geq 0$

تكون $1 = \text{sgn}(a)$ إذا كانت $a < 0$

يرمز إلى دالة أكبر عدد صحيح بالرمز $[a]$ وهى تعطى أكبر عدد صحيح أقل أو يساوى a

جدول أ – ١

الوصف الكلامى	التعريف	اسم الدالة بالخلاصات	نوع الدالة
جذر تربيعى	$\sqrt{a}$	SQRT(X) DSQRT(DX) CSQRT(C)	حقيق دقة متضاعفة مركب
أس	e^a	EXP(X) DEXP(DX) CEXP(C)	حقيق دقة متضاعفة مركب
لوغاريتم طبيعى (للأساس ه)	$\log_e(a)$	ALOG(X) DLOG(DX) CLOG(C)	حقيق دقة متضاعفة مركب
لوغاريتم عادى (للأساس 10)	$\log_{10}(a)$	ALOG10(X) DLOG10(DX)	حقيق دقة متضاعفة
قيمة مطلقة	$\|a\|$	IABS(J) ABS(X) DABS(DX)	صحيح حقيق دقة متضاعفة
دوال ذات خلاصة مركبة $C = x + iy$ القيمة مرافق مركب جزء حقيق جزء تخيل	 $\sqrt{x^2 + y^2}$ $x - iy$ x y	 CABS(C) CONJG(C) REAL(C) AIMAG(C)	 حقيق مركب حقيق حقيق

جدول أ – ١ (تابع)

الوصف الكلامى	التعريف	اسم الدالة بالخلاصات	نوع الدالة
الشكل المركب لرقين حقيقين (x, y)	$x + iy$	CMPLX(X, Y)	مركب
تحويل النوع	تحويل صحيح إلى حقيقى	FLOAT(J)	حقيقى
	تحويل صحيح إلى دقة متضاعفة	DFLOAT(J)	دقة متضاعفة
	تحويل حقيقى إلى دقة متضاعفة	DBLE(X)	دقة متضاعفة
	تحويل دقة متضاعفة إلى حقيقى بالبتر	SNGL(DX)	حقيقى
	تحويل حقيقى إلى صحيح بالبتر	IFIX(X)	صحيح
أكبر قيمة	$\text{Max}(a_1, a_2, \ldots)$	MAX0(J, K, . . .)	صحيح
		MAX1(X, Y, . . .)	صحيح
		AMAX0(J, K, . . .)	حقيقى
		AMAX1(X, Y, . . .)	حقيقى
		DMAX1(DX, DY, . . .)	دقة متضاعفة
أصغر قيمة	$\text{Min}(a_1, a_2, \ldots)$	MIN0(J, K, . . .)	صحيح
		MIN1(X, Y, . . .)	صحيح
		AMIN0(J, K, . . .)	حقيقى
		AMIN1(X, Y, . . .)	حقيقى
		DMIN1(DX, DY, . . .)	دقة متضاعفة
الفرق الموجب أو صفر ما بين (a_1, a_2)	$a_1 - \min(a_1, a_2)$	IDIM(J, K)	صحيح
		DIM(X, Y)	حقيقى
الدوال المثلثية لها خلاصات التقدير الدائرى	$\sin(a)$	SIN(X)	حقيقى
		DSIN(DX)	دقة متضاعفة
		CSIN(C)	مركب
	cos	COS(X)	حقيقى
		DCOS(DX)	دقة متضاعفة
		CCOS(C)	مركب
	$\tan(a)$	TAN(X)	حقيقى
		DTAN(DX)	دقة متضاعفة
نقل العلامة ما بين (a_1, a_2)	$\lvert a_1\rvert \cdot \text{sgn}(a_2)$	ISIGN(J, K)	صحيح
		SIGN(X, Y)	حقيقى
		DSIGN(DX, DY)	دقة متضاعفة
بتر الحدد للمصفوفة	$\text{sgn}(a) \cdot [\lvert a\rvert]$	INT(X)	صحيح
		AINT(X)	حقيقى
		IDINT(DX)	صحيح
إيجاد المثل ذى القيمة الصغرى المطلقة فى قسم الباقى a_1 (modulo a_2)	$a_1 - \text{sgn}(b)[\lvert b\rvert] \cdot a_2$, where $b = a_1/a_2$	MOD(J, K)	صحيح
		AMOD(X, Y)	حقيقى
		DMOD(DX, DY)	دقة متضاعفة

ملحق (ب)

التمثيل الداخلى للبيانات

مقدمة

تتكون كل خلية ذاكرة من نبائط التوازن ، أى ، نبائط لها حالتى اتزان وتمثل هاتين الحالتين بالأرقام الثنائية 0 و 1 أى تستطيع كل نبيطة ثنائية التوازن تخزين بيت (*bit*) (اختصار رقم ثنائى binary digit) من المعلومات . وتستخدم عادة مجموعة أو سلسلة من الأرقام الثنائية لتمثيل وحدة من المعلومات .

حيث يمكن أن ترى سلاسل من الخانات الثنائية كأنها أرقام ثنائية ، إن تمثيل الأرقام فى النظام الثنائى هام فى علوم الحاسب . يبين الجدول ب — ١ التمثيل الثنائى لأول ستة عشر رقماً صحيحاً موجباً . يستخدم الدليل 2 أحياناً لتمييز الرقم الثنائى عن الرقم العشرى ، فمثلا :

$(101101)_2$ تعنى الرقم الثنائى 101101

ذو التمثيل العشرى 45 . وتقع دراسة التحويل من الأرقام العشرية إلى الأرقام الثنائية وبالعكس خارج نطاق هذا الملحق .

جدول ب — ١

رقم ثنائى	رقم عشرى
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16

كلمات وبايتات

تسمى الوحدة الأساسية للمعلومات التى تخزن أو تسترجع من خلية ذاكرة بكلمة (*word*) . يمكن أن تتكون الكلمة من بعض الأحجام الشائعة الكلمة 24 أو 30 أو 32 و 36 و 48 و 60 موضع ثنائى . هناك حاسبات بكلمات ذات طول ثابت وحاسبات بكلمات ذات طول متغير . ومع ذلك فكل كلمة فى الذاكرة يجب أن يكون لها عنوانها الخاص .

وعلاوة على ذلك ، كلما أدخلنا أو أخرجنا كلمة من الذاكرة ، فإن كل مواضعها الثنائية تنقل فى آن واحد .

تسمح بعض الحاسبات أيضاً بتخزين أو استرجاع جزء من الكلمة من الذاكرة . تسمى هذا الجزء من الكلمة باسم بايت . وتتكون البايت عادة من 6 أو 8 مواضع ثنائية ويمكن أن تأخذ عنواناً منفصلا . يطلق على هذه الآلات بأن لها خصائص المعنونة بالبايت (*byte addressable*) . وسلسلة CDC 6000 — 7000 بها كلمات تتكون من عشرة بايتات كل ذات ستة مواضع ثنائية وبهذا تتكون الكلمة من 60 موضعاً ثنائياً . لأغراض التوضيح ، سنعتبر أن آلتنا الحاسبة تشبه سلسلة IBM 360/370 والتى لها كلمات مكونة من أربعة بايتات كل ذات 8 مواضع ثنائية . وبهذا تتكون الكلمة من 32 موضعاً ثنائياً كالتالى :

كلمة

Byte 0	Byte 1	Byte 2	Byte 3

وإن الكلمة كوحده للمعلومات هامة أيضاً للآلات byte-addressable (المعنونة بالبايت) وتعطى عنوان البايت أقصى اليسار .

بيانات حرفية

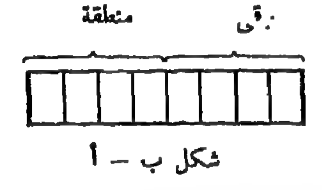
يستطيع أى مترجم فورتران عادة أن يتعرف على الحروف وعددها 48 وهى مذكورة فى الفصل الثانى ، بالتحديد 10 أرقام و26 حرفاً أبجدياً و 12 حرفاً خاصاً . حيث

$$2^5 < 48 < 2^6$$

فيكون المطلوب على الأقل 6 مواضع ثنائية لتمثيل هذه الحروف . يمكن لأى كود حرف ذى 6 مواضع ثنائية أن يمثل حتى 64 حرفاً مختلفاً مما يسمح بعدد 28 = 36 — 64 حرفاً خاصاً كحد أقصى . وتستخدم معظم حاسبات الجيل الثالث كود حروف ذات 8 — مواضع ثنائية وهذا يسمح بعدد يصل إلى 256 حرف وهو أكثر من كاف .

تكون عادة الحروف الأبجدية فى ترتيبها الأ ، أى كود الحرف A أصغر من كود الحرف B ، وهكذا . يشكل هذا التكويد سلسلة مفروزة يمكن أن تستخدم نات الأبجدية . (انظر الفصل التاسع) .

يرتب عادة كود 8 موضع ثنائى (6 ثنائى) فى مناطق بحيث يكون الكود المقابل لحرف معين فى الأربع (الثلاثة) مواضع الثنائية أقصى اليسار ، والقيم الرقية ربعة (الثلاثة) مواضع الثنائية أقصى اليمين . (انظر شكل ب — ١) . ويسمى كود الحرف المستخدم بالآلات 60/370 II كود EBCDIC أى (نظام ترميز ممتد للتبديل العشرى مكود ثنائياً)



شكل ب — ١

ويبين جدول ب — ٢ الكود EBCDIC للحروف الأبجدية (Extended Binary Coded Decimal Interchange Code) والأرقام (10) (62) لاحظ أن الكود بالنسبة لأى رقم يحتوى 1111 فى قسم المناطق وتمثيله الثنائى فى القسم الرقى . ولما كان من غير الممكن تخزين أكثر من أربعة حروف ذات 8 — مواضع ثنائية فى كلمة ذات 32 موضع ثنائى ، فإن هذا يفسر السبب فى أن سعة الحرف فى سلسلة IBM 360/4370 هى M = 4

جدول ب – ٢ أكواد EBCDIC

حرف	EBCDIC	حرف	EBCDIC	حرف	EBCDIC	حرف	EBCDIC
	منطقة أرقام		منطقة أرقام		منطقة أرقام		منطقة أرقام
						0	1111 0000
A	1100 0001	J	1101 0001			1	0001
B	0010	K	0010	S	1110 0010	2	0010
C	0011	L	0011	T	0011	3	0011
D	0100	M	0100	U	0100	4	0100
E	0101	N	0101	V	0101	5	0101
F	0110	O	0110	W	0110	6	0110
G	0111	P	0111	X	0111	7	0111
H	1000	Q	1000	Y	1000	8	1000
I	1100 1001	R	1101 1001	Z	1110 1001	9	1111 1001

بيانات رقمية

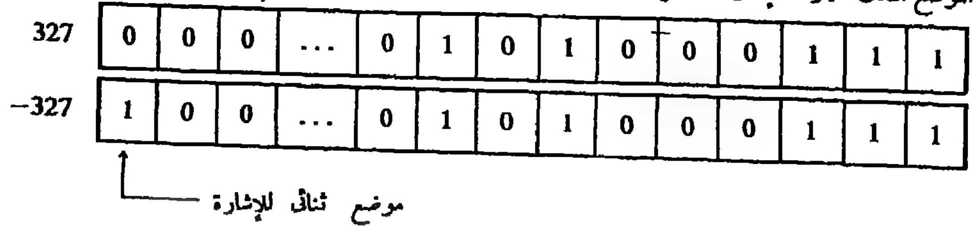
هناك طريقتان مختلفتان لتكويد بيانات رقمية . أما تكويد الكمية أو تكويد الأرقام المفردة . نناقش أولا تكويد الكمية .

تكويد الكمية

يجرى الفورتران كلا من حسابات صحيحة ونقطة عائمة للحساب ويختلف التمثيل الداخلى لهذين النوعين من البيانات الرقمية :

(ا) أعداد صحيحة

الطريقة المباشرة لتمثيل عدد صحيح فى خلية ذاكرة تكون عن طريق التمثيل الثنائى . بفرض أن خلايا الذاكرة ذات 32 موضع ثنائى ، يترك الموضع الثنائى الأول للإشارة ، تخزن الأعداد الصحيحة 327 و 327 . كالتالى :



وبالتالى ، فأكبر رقم يمكن أن يخزن فى 32 خانة ثنائية هو

$$0(\underbrace{111\ldots1111}_{31\ \text{ones}})_2$$

ويمثل (2^{31} — ١) ويترتب على أى محاولة لتخزين أعداد صحيحة أكبر من السعة حدوث طفح ومن المحتمل أن تفقد الأرقام المئوية قصوى .

لأسباب حسابية تخزن الأرقام السالبة فى الذاكرة باستخدام ما يسمى بالتمثيل المكمل للأساس ناقص واحد ***radix-minus-one*** أو المكمل للأساس (*radix*) سنناقش أولا هذه التمثيلات فى النظام العشرى حيث يطلق عليهم المكمل التاسع (*nine's*) والمكمل العاشر (*ten's*) على الترتيب . نحصل على المكمل التاسع (nine's) لرقم عشرى بطرح كل رقم عشرى من 9 والمكمل العاشر (ten's) لرقم هو مكمله التاسع (nine's) مضافاً إليه واحد .

على سبيل المثال :

رقم عشرى	320	123	327
مكمل التسعات (nine's)	679	876	672
مكمل العشرات (ten's)	680	877	673

تسمى المكملات المناظرة للأرقام الثنائية المكمل الأحادى (*one's*) والمكمل الثنائى (*two's*) على الترتيب . بالمثل ، نحصل على المكمل الأحادى (one's) لرقم ثنائى بطرح كل خانة ثنائية من 1 (أى ، تحويل كل 0 إلى 1 وكل 1 إلى 0) والمكمل الثنائى (two's لرقم عشرى هو مكملة الأحادى (one's) مضافاً إليه واحد . فمثلا .

رقم ثنائى	10100	10111	11001
المكمل الاحادى (one's)	01011	01000	00110
المكمل الثنائى (tw's)	01100	01001	00111

يسمح لنا تمثيل الأرقام السالبة بالمكمل بإجراء الطرح باستخدام الجمع . (وحيث أن الضرب هو جمع متكرر والقسمة هى طرح متكرر ، بذلك يمكن إجراء كل الحسابات الصحيحة باستخدام الجمع) . نوضح كيف يعمل هذا الحساب بمثال .

نظام ثنائى يستخدم المكمل الأحادى (one's) ذو كلمة بها 32 موضع ثنائى .	نظام عشرى يستخدم المكمل التاسع (nine's)	حساب عادى
00000 . . . 0001011000001	705	705
+11111 . . . 1111010111000	+672	−327
(1)00000 . . . 0000101111001	(1)377	378
→ 1	→ 1	
00000 . . . 0000101111010	378	

لاحظ أن 1 الموجود على أقصى اليسار يحذف ثم يضاف إلى المجموع ، كما هو موضح بالأسهم — يسمى هذا بالترحيل عبر النهاية (end-around carry) وتفاصيل هذا الحساب تقع خارج نطاق هذا الملحق ولذا نكتفى بهذا القدر من الشرح .

(ب) أرقام بنقطة عائمة (ذات علامة عشرية)

تعطى دائما الجمع والطرح والضرب الصحيح نتائج صحيحة . ومع ذلك فالقسمة الصحيحة تعطى أيضاً نتيجة صحيحة لكن جزءها العشرى لن تخزن قيمته ، وعلاوة على ذلك لا يمكن أن يخزن رقم ذى قيمة مطلقة تتجاوز قيمته 2^{31} فى كلمة ذات 32 موضع ثنائى . هذه القيود هى السبب الأساسى لاستخدام أرقام بنقطة واحدة .

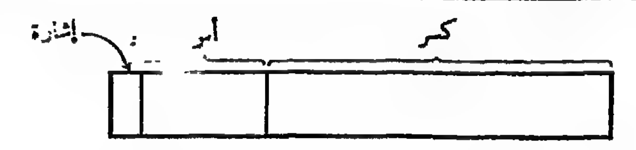
اعتبر الرقم 32.75 — يمكن أن يكتب باستخدام ترميز أسى قياسى كالآتى :

$$-0.3275 \times 10^2 \text{ مع } \begin{cases} - \\ 0.3275 \\ 2 \text{ (للأساس 10)} \end{cases}$$

يمكن أن يستخدم تمثيل مشابه للأرقام الثنائية . فمثلا ، يمكن أن يكتب الرقم العشرى 23.75 — كرقم ثنائى $(100000.11)_2$ ويوضع فى الشكل القياسى :

$$-0.10000011 \times 2^6 \text{ مع } \begin{cases} \text{إشارة} & - \\ \text{كسر} & 0.10000011 \\ \text{أس} & 6 \text{ (أساس 2)} \end{cases}$$

واضح أنه يمكن وضع كل رقم حقيقى فى الشكل القياسى لنحصل على إشارة وحيدة ، كسر ثنائى وحيد ، وأس ثنائى وحيد ، بذلك يمكن أن يخزن داخلياً كما يلى :



في الكلمة ذات 32 موضع ثنائي ، نحتفظ بموضع ثنائي واحد للإشارة و ٧ مواضع ثنائية للأس ، ومن ثم ، عدد $24 = 8 - 32$ موضع ثنائي للكسر الثنائي . يجدر ملاحظة أن أي رقم حقيقي ، سواء خزن باستخدام حقل [illegible] F- له نفس التمثيل الداخلي كما هو موضح عاليه .

وحيث أن الأس يمكن أن يكون سالباً ، ويجب أن تعمل بعض التنظيمات لإشارته بداخل [illegible] بدلا من استخدام موضع ثنائي واحد لتعريف إشارة الأس . ويتم ذلك بتحويل الأس الحقيقي إلى ما يسمى بـ [illegible] (characteristic) [illegible] $64 = 2^6 = (100000)$ وبعد ذلك يخزن المميز بدلا من الأس الحقيقي . وفيما يلي العلاقة بين الأس الحقيقي والمميز .

الأس الحقيقي	63	...	1	0	−1	...	−61	−62	−63	−64
المميز	127	...	65	64	63	...	3	2	1	0

(في حالة استخدام حقل ذي 8 مواضع ثنائية للأس ، نجمع $128 = 2^7$ إلى الأس الحقيقي لنحصل على المميز الخاص به . وهكذا) .

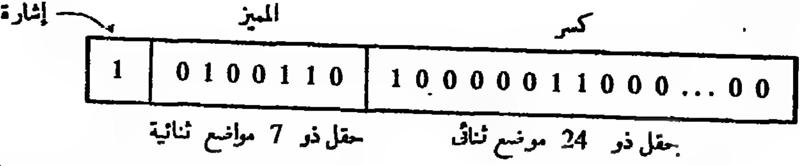
[illegible] 32.75 — مرة أخرى ، حيث تمثيله الثنائي القياسي هو

$$-0.10000011 \times 2^6$$

الأس الحقيقي هنا هو [illegible] يكون المميز هو

$$6 + 64 = 70 = (100110)_2$$

وبذلك يخزن 32.75 — داخليا كما يلي :



نستطيع أن نرى أن أكبر رقم بالنقطة العائمة الذي يمكن أن يخزن في كلمة ذات 32 موضع ثنائي (باستخدام الشكل الثنائي ذو النقطة العائمة) هو 2^{63} وأصغر رقم هو 2^{-64} أي محاولة لتخزين أعداد أكبر من 2^{63} (أو أصغر من 2^{-64}) ستتسبب في طفح في النقطة العائمة (أو قحط) . ويسبب حجم الكلمة المحدود تمثيلا داخلياً غير دقيق للأرقام بالنقطة العائمة .

أخيرا ، تذكر أن في شكل النقطة العائمة ، لا يستخدم ترميز المكمل للكسور السالبة .

تكويد الخانات المفردة

وسيلة أخرى لتكويد البيانات الرقمية هي استخدام كود الحرف (الصيغة ذات المناطق) لكل خانة من خاناته . في مثل هذه الحالة تخصص الإشارة إلى حقل المنطقة (الخانة) على أقصى اليمين كما يلي :

منطقة	خانة	zone	digit	...	zone	digit	sign	digit

في EBCDIC تمثل إشارة الزائد والناقص بـ 1100 و 1101 على الترتيب . بكود الأرقام 327 و 327 — استخدام جدول ب — ٢ لأكواد الأرقام كما يلي :

7	−	2		3	
0111	1101	0010	1111	0011	1111

7	+	2		3	
0111	1100	0010	1111	0011	1111

المذكور أعلاه هو الطريقة التى تمثل بها البيانات الرقية فى الإدخال والأخراج ومع ذلك فعندما تستخدم البيانات الرقية فى الحسابات ، فإنها تحول إلى شكل مختلف حتى يتم تشغيلها بسهولة . أحد هذه الأشكال هو النظام العشرى المكود ثنائياً **(BCD) (Binary Coded Decimal)**

يشبه التكويد BCD للأرقام ، التكويد المذكور أعلاه فيما عدا استخدام حقول المنطقة ويظهر كود الإشارة فى الحقل أقصى اليمين . بمعنى آخر ، فإننا نستخدم تمثيل ذى 4 مواضع ثنائية لكل رقم عشرى فى العدد مع كود الإشارة فى النهاية . بالتالى ، تكود 327 و 327— فى BCD كما يلى :

−	7	2	3
1101	0111	0010	0011

+	7	2	3
1100	0111	0010	0011

تتم عملية الجمع فى BCD بالجمع الثنائى للأرقام العشرية المفردة ، ومع ذلك ، عندما يتعدى المجموع تسعة ، يرحل وحده إلى المكان التالى كما فى الحساب العشرى العادى . وأى ، مناقشة أخرى لهذا الحساب تكون خارج نطاق هذا الملحق .

المصطلحات العلمية (عربى ــ انجليزى)

(ا)

ابجدى رقمى	Alphanumeric
اختبار الصدى	Echo check
ادخال/اخراج	INPUT/OUTPUT
ادخال/اخراج بصياغة طليقة	Format-free Input/Output
ادماج	Merging
ازالة الخطأ	Debugging
اســـاس	Radix
استدعاء	Invoke
اســـم	Name
اسى	Exponential
آلـــة	Machine
اماكن التخزين	Storage locations
انتقال غير مشروط	Unconditional transfer
انتقال مشروط	Conditional transfer
انتقاء	Selection

(ب)

بايت (مجموعة خانات)	Byte
بحث ثنائى	3inary search
بديـــل	ılternative
برامج فرعية	ubprograms
برامج مخزونة	red programs
برنامج المصدر	rce program
برنامج الهدف	ct program
بطاقة	'd

Control card	بطاقة التحكم
Comment card	بطاقة تعليق
Sentinel card	بطاقة حارسة
Trailer card	بطاقة خلفية
Header card	بطاقة المقدمة
End-of-file card	بطاقة نهاية الملف
Dimension	بعــد
Data	بيانات

(ت)

Keypunching	تثقيب
Carriage control	تحكم العربة
Conversion	تحويــل
Storing numbers	تخزين الأعداد
Assignment	تخصيص
Hierarchy	تدرج هرمى
Binary coded decimal	ترميز عشرى مكتوب بالرمز الثنائى
Left-justified	تضبيط جهة اليسار
Right-justified	تضبيط جهة اليمين
Parenthesis-free expression	تعبير خال من الاقواس
Arithmetic expression	تعبير رياضى
Relational expression	تعبير مترابط
Logical expression	تعبير منطقى
Back-substitution	تعويض خلفى
Round-off	تقــريب
Programming technique	تقنية البرمجة
Numerical integration	تكامل عددى
Graphing	تمثيل بيانى
Internal representation	تمثيل داخلى
Distribution, frequency	توزيع ، تردد (تكرار)
Interval halving	تنصيف الفترات

(ث)

ثـــابت	Constant
ثنـــائى	Binary

(ج)

جدول الأسبقيات	Precedence table
جذور المعادلة	Roots of equation
جـــرد	Inventory
جزء عشرى	Mantissa
جمــلة	Statement
جهاز (وحدة)	Device

(ح)

حاسب	Computer
حرف / حرفى	Character/literal
حزمـــة	Package
حساب	Arithmetic
حقـــل	Field
حـــلقة	Loop

(خ)

خـــالية	Blank
خروج من	Exit from
خريطة سير العمليات	Flowchart
خطـــأ	Error
خطوة التزايد	Increment
خطى	Linear
خلاصة	Argument
خوارزم (نظام حسابى)	Algorithm

(د)

Function	دالة
Subscript	دليل

(ذ)

Memory	ذاكرة
Two-dimensional	ذو بعدين

(ر)

Binary digit	رقم ثنائى
Symbol	رمز

(ز)

Dummy	زائف

(س)

Record	سجل
Capacity	سعة
String	سلسلة من الحروف

(ش)

Form	شكل

(ص)

Integer	صحيح
Format	صيغة
Execution-time format	صيغة زمن التنفيذ

(ض)

Implied	ضمنى

(ط)

Method	طريقة

(ع)

Number	عدد
Prime number	عدد أولى
Real number	عدد حقيقى
Width	عــرض
Decimal	عشرى
Operations	عمليات
Column	عمــود
Element	عنصر
Return	عودة

(غ)

Unexecutable	غير قابلة للتنفيذ
Unformatted	غير مصاغ

(ف)

Exchange sort	فرز تبادلى
Internal sort	فرز داخلى
Bubble sort	فرز فقاعى

(ق)

Executable	قابلة للتنفيذ
Decision	قــرار
Read-in	قراءة داخلية
Initial value	قيمة ابتدائية
Test value	قيمة الاختبار
Limiting value	قيمة محدودة
End value	قيمة النهاية

(ك)

كتلة	Block
كفاءة	Efficiency
كلمة	Word
كمية غير متجهة	Scalar
كود	Code

(ل)

لا نهائية	Infinite
لغات برمجة	Programming language
لغة التجميع	Assembly language
لغة رفيعة المستوى	High level language

(م)

متتابع	Sequential
متداخلة	Nested
مترابط	Relational
مترجم	Compiler
متعدد الأبعاد	Multidimensional
متعددة	Multiple
متسلسل	Sequence
متضاعف الدقة	Double precision
متغير	Variable
مجرى	Stream
مجمع	Assembler
مجموعة	Set
مجموعة بطاقات	Deck of cards
مجموعة متراصة	Array
مخرج (خروج) غير طبيعى	Abnormal exit
مدخل	Entry
مركب	Complex
مشروط	Conditional

Operator	مشـغـل
Formatted	مصـــاغ
Matrix	مصفوفة
Equation	معـــادلة
Binomial coefficient	معامل ذو حدين
Generalized	معممة
Labeled	معنونة
Scale, factor	مقياس ، معامل
Complement	مكمـــل
Characteristic	خاصية
Logical	منطقى
Single precision	منفرد الدقة
Specifications	مواصـــفات

(ن)

Fixed point	نقطة ثابتة
Floating point	نقطة طليقة (عائمة)
Transfer of control	نقـــل التحكم
Mode of operation	نمط التشـغيل
Mixed mode	نمط مختـــلط

(هـ)

Destructive	هدامة
Structure	هيكل

(و)

Unit	وحدة
Input unit	وحدة الادخال
Control unit	وحدة التحـــكم
Central processing unit (CPU)	وحدة التشـغيل المركزية

المصطلحات العلمية (انجليزى ــ عربى)

(A)

Abnormal	غير طبيعى
Address	عنـــوان
Algorithm	خوارزم (نظام حسـابى)
Alphanumeric	ابجدى رقمى
Alternative	بديـــل
Argument	خلاصة
Arithmetic expression	تعبير رياضى
Array	مجموعة متراصة
Assembler	مجمـــع
Assembly language	لغة التجميع
Assignment	تخصيص
Augmented	مـــزادة

(B)

Back substitution	تعويض خلفى
Binary	ثنـائى
Binary coded decimal (BCD)	ترميز عشرى مكتوب بالرمز الثنائى
Binary digit	رقم ثنائى
Binomial	ذو حدين
Blank	خالية
Block	كتـلة
Bubble sort	فرز فقاعى
Byte	بايت (مجموعة خانات)

(C)

Capacity	سـعة

Card	بطاقة
Carriage control	تحكم العربة
Central processing unit (CPU)	وحدة التشغيل المركزية
Character	حــرف
Characteristic	خاصية
Code	كــود
Column	عمــود
Comment	تعليق
Compiler	مترجم
Complement	مكمــل
Complex	مركب
Computer	حاسب
Conditional transfer	انتقال مشروط
Constant	ثــابت
Control	تحــكم
Control unit	وحدة التحكم
Conversion	تحويــل

(D)

Data	بيــانات
Debugging	ازالة الخطأ
Decimal	عشرى
Decision	قــرار
Deck of cards	مجموعة بطاقات
Destructive	هــدامة
Device	جهاز (وحدة)
Dimension	بعــد
Distribution	توزيع
Double precision	متضاعف الدقة
Dummy	زائــف

(E)

Echo check	اختبار الصدى
Efficiency	كفــاءة
Element	عنصر
End-of-file	نهاية الملف
End value	قيمة النهاية
Entry	مدخــل
Equation	معادلة
Error	خطــأ
Exchange	تبــادل
Executable	قابلة للتنفيذ
Execution-time	زمن التنفيذ
Exit from	خروج من (خرج)
Exponential	اسى
Expression	تعبير

(F)

Factor	معامل
Field	حقل
File	ملف
Fixed point	نقطة ثابتة
Floating point	نقطة عائمة (طليقة)
Flowchart	خريطة سير العمليات
Form	شــكل
Format	صــيغة
Format-free	بدون صياغة
Frequency	تردد (تكرار)
Function	دالــة

(G)

Generalized	معممة
Graphing	تمثيل بيانى

(H)

Header card	بطاقة المقدمة
Hierarchy	تدرج هرمى
High level language	لغة رفيعة المستوى

(I)

Implied	ضـــمنى
Increment	خطوة التزايد
Input-output	ادخال/اخراج
Integer	صحيح
Interval halving	ـيف الفترات
Inventory	جرد ــ قائمة المخزون
Invoke	استدعاء

(K)

Keypunching	تثقيب

(L)

Label	عنـــوان
Left-justified	مضبط جهة اليسار
Limiting value	قيمة محددة
Linear	خطى
Literal	حرفى
Logical	منطقى

(M)

Machine	آلـــة
Mantissa	جـــزء عشرى
Matrix	مصفوفة
Memory	ذاكرة
Merging	ادماج
Method	طريقة

Mixed mode	نمط مختلط
Mode of Operation	نمط التشغيل
Multidimensional	متعدد الأبعاد
Multiple	متعدد (مضاعفات)

(N)

Name	اسم
Nested	متداخلة
Number	عدد
Numerical integration	تكامل عددى

(O)

Object program	برنامج الهدف
Operator	مشغل

(P)

Package	حزمة
Parenthesis-free	خال من الأقواس
Precedence table	جدول الاسبقيات
Prime number	عدد أولى
Programming languages	لغات برمجة
Programming technique	تقنية البرمجة

(R)

Radix	أساس
Record	سجل
Relational	مترابط
Return	عودة
Right-justified	مضبط جهة اليمين
Roots of equation	جذور المعادلة
Round-off	تقريب

(S)

Scale	مقياس
Scalar	كمية غير متجهة
Selection	انتقـــاء
Sentinel card	بطاقة حارسة
Sequence	تسلسل
Sequential	متتابع
Set	مجموعة
Single precision	منفرد الدقة
Source program	برنامج المصدر
Specifications	مواصفات
Statement	جملة
Storage locations	اماكن التخزين
Stored programs	برامج مخزونة
Storing numbers	تخزين الأعداد
Stream	مجرى
String	سلسلة من الحروف
Structure	هيكل
Subprograms	برامج فرعية
Subscript	دليل
Symbol	رمـــز

(T)

Test	اختبار
Trailer card	بطاقة خلفية
Transfer of control	نقل التحكم
Two-dimensional	ذو بعدين

(U)

Unconditional	غير مشروط

Unexecutable غير قابلة للتنفيذ

Unit وحـــدة

(V)

Variable متغير

(W)

Width عــرض

Word كلمة

الفهرس الأبجدى



مطابع الـدار الهندسـية



PROGRAMMING WITH FORTRAN

Including structured Fortran

Lipschutz

صـــدر أيضـــاً للنـــاشر
فى الحاسبـــــات

★ المدخل لعلم الحاسبات	BARTEE
★ البرمجة بلغة البيسك – شوم	GOTTFRIED
★ البرمجة بالفورتران – شوم	LIPSCHUTZ
★ الرياضيات الأساسية للحاسب – شوم	LIPSCHUTZ
★ المجهزات والحاسبات الدقيقة لطلبة الهندسة والفنيين	WOOLLARD
★ الدوائر المتكاملة واستخدامات المجهزات الدقيقة	MORRIS
★ الدوائر المتكاملة الرقمية والحاسبات	WOOLLARD
★ البرمجة بلغة البيسك – إنجليزى	GOTTFRIED

الدار الدولية للنشر والتوزيع

[illegible]

[illegible] – القاهرة

[illegible]

[illegible]

07 084821 1